AD-A079 316 MICHIGAN UNIV ANN ARBOR DEPT OF NAVAL ARCHITECTURE --ETC F/G 20/4
HEADSEAS WAVE DIFFRACTION COMPUTER PROGRAM. USER MANUAL.(U)
AUG 79 R F BECK N00014-78-C-0109

UNCLASSIFIED NL

1 OF 2
AD
A079316

6

USER MANUAL FOR HEADSEAS WAVE DIFFRACTION COMPUTER PROGRAM

Robert F. Beck

11

Aug[redacted]79

12 108

This research was carried out under the
Naval Sea Systems Command
General Hydromechanics Research Program
Subproject SR 023 01 01, administered by the
Naval Ship Research and Development Center
Contract No. N00014-78-C-0109

15

Department of Naval Architecture
and Marine Engineering
College of Engineering
The University of Michigan
Ann Arbor, Michigan 48109

401 188

DISCLAIMER NOTICE

TABLE OF CONTENTS

INTRODUCTION

The purpose of this computer program is to compute the dynamic pressure distribution and related quantities of interest due to the diffraction of sinusoidal head waves. The method of computation is based on slender-body theory. The theoretical analysis is based on the assumption that the ship is slender. In addition, it is assumed that the incident waves are of small amplitude and their wavelength is short relative to the ship length. In the next section we shall give a brief summary of the theoretical analysis in order to facilitate an understanding of the computer program. Details of the theoretical analysis may be found in Beck (1979). In the following sections, details of the numerical technique and the computer program will be discussed.

THEORETICAL ANALYSIS

The coordinate system is shown in Figure 1. The origin is at the bow with the x-axis pointing aft. The z-axis is vertical upward and the x-y plane

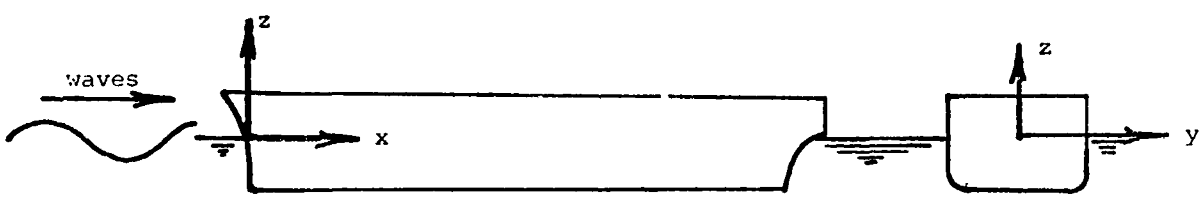

Figure 1. Coordinate System

is coincident with the calm-water plane. In this coordinate system, the incident wave potential is given by

$$\phi_I(x,y,z)e^{i\omega_e t} \tag{1}$$

where

$$\phi_I = \frac{ga}{\omega_o} e^{\nu z} e^{-i\nu x}$$

g = acceleration gravity

a = wave amplitude

ν = wave number

$= \omega_0^2/g$

U = forward speed

ω_o = absolute wave frequency

ω_e = frequency of encounter

$= \omega_0 + \omega_0^2 U/g$

The diffraction potential is written as

$$\phi_D(x,y,z)e^{i\omega_e t} \tag{2}$$

To find ϕ_D we must solve both the near-field and far-field problems. In the far-field, the solution is represented by a line distribution of pulsating sources, the strength of which is given by

$$\sigma(x)e^{i(\omega_e t - \nu x)}$$

The source strength, $\sigma(x)$, is found by solving the Volterra integral equation

$$\frac{ga}{\omega_0} + \sigma(x)\left[\frac{1}{\pi B_o(x)} - C\right] - \alpha\int_0^x d\xi \frac{\sigma(\xi)}{\sqrt{x-\xi}} = 0 \tag{3}$$

where

$$C = \begin{cases} -i/2 & \tau = 0 \\ \dfrac{1}{2\sqrt{2}} & \tau >> 1/4 \end{cases}$$

$$\alpha = \sqrt{\frac{\nu}{2\pi(1+2\tau^*)}}\; e^{-i\pi/4}$$

$$\tau = \frac{\omega_e U}{g}$$

$$\tau^* = \frac{\omega_o U}{g}$$

$B_0(x)$ = near field source strength.

The near-field source strength is found by solving the near field problem and may be assumed known when solving equation (3).

In the near-field, the diffraction potential is written as

$$\Phi_D = \phi(x,y,z)e^{-i\nu x} \tag{4}$$

Both the first and second order near-field solutions must be determined. The first order solution is simply the negative of the incident wave. The second-order solution must satisfy the Helmholtz equation subject to boundary conditions on the free surface and the body surface. At infinity, the near-field solution must match the far-field solution.

As discussed in Beck (1979), the two-term near-field solution can be written as

$$\phi(x,y,z) \sim -\frac{ga}{\omega_0} e^{\nu z} + A(x)\left[e^{\nu z}+B_0(x)S(y,z) + \sum_{n=1}^{\infty} B_n(x)O_n(y,z)\right] \tag{5}$$

The coefficient $A(x)$ is determined by matching with the far-field solution and we find

$$A(x) = -\frac{\sigma(x)}{\pi B_0(x)} \tag{6}$$

The terms inside the square brackets represent Ursell's (1968a) solution to the Helmholtz equation subject to the free surface and body boundary conditions. $S(y,z)$ is a source-like term and O_n are wave-free potentials. The coefficients B_0, B_n are determined by satisfying the body boundary condition, which, in this case, is obtained by setting the normal derivatives of the terms in square brackets equal to zero on the body surface.[1]

The determination of the coefficients B_0, B_n for arbitrary body shapes is often very difficult. Troesch (1976) has avoided this problem by using an integral-equation technique. Troesch's technique and computer program were

1. This type of solution is often called a multi-pole expansion.

actually developed for the solution of the oblique seas case, but they can be modified for use in the present problem. We first write Ursell's solution in the form

$$\psi(y,z;x) = e^{+\nu z} + \Phi(y,z;x) \tag{7}$$

The potential $\Phi(y,z,x)$ must satisfy the Helmholtz equation

$$\frac{\partial^2\Phi}{\partial y^2} + \frac{\partial^2\Phi}{\partial z^2} - \nu^2\Phi = 0 \tag{8}$$

subject to boundary conditions on the free surface and the body. At infinity, the behavior of Ψ must match the far-field solution. The free-surface boundary condition is

$$\frac{\partial\Phi}{\partial z} - \nu\Phi = 0 \qquad \text{on } z=0$$

On the body, the boundary condition is

$$\frac{\partial}{\partial N}\left[e^{\nu z} + \Phi(z,y;x)\right] = 0$$

or

$$\frac{\partial\Phi}{\partial N} = -\frac{\partial e^{\nu z}}{\partial N} \qquad \text{on } h(y,z;x) = 0 \tag{9}$$

where

$h(y,z,x)$ = equation of body surface in y-x plane

$\underline{N}$ = two-dimensional unit normal to body surface in the y-z plane $= (N_2, N_3)$

The positive sense of $\underline{N}$ is into the body.

The problem for the potential Φ is now equivalent to the boundary value problem solved by Troesch. He writes the solution as a distribution over the body surface of two-dimensional sources. Thus, we have

$$\Phi(y,z;x) = \int_{C(x)} dl\ \gamma(\eta,\zeta)G(y,z;\eta,\zeta) \tag{10}$$

where the line integral is taken along the body contour. $\gamma(\eta,\zeta)$ is the two-dimensional source strength, which is determined by satisfying the body boundary condition. $G(y,z;\eta,\zeta)$ is the Green function, which satisfies the Helmholtz equation and the free surface boundary condition (for details see Ursell (1968b)). By matching the limiting values of both the multipole solution and the Green function solution for large values of y, it can be shown that

$$B_O(x) = 2 \int_{C(x)} dl \; \gamma(\eta,\zeta) e^{\nu\zeta} \tag{11}$$

Equation (11) allows the determination of the near-field source strength, $B_O(x)$, by using the integral-equation solution technique without developing the multipole expansion.

The pressure acting on the body is found using the linearized Bernoulli equation, which may be written as

$$\begin{aligned} \bar{p} &= pe^{i\omega_e t} \\ &= -\rho\left(i\omega_e + U \frac{\partial}{\partial x}\right)(\phi_I+\phi_D) e^{i\omega_e t} \end{aligned} \tag{12}$$

Substituting the expressions for ϕ_I and ϕ_D into equation (12) and retaining only the lowest-order terms (see Beck (1979)), we find the following expression for the nondimensional pressure amplitude:

$$\begin{aligned} p^* &= \frac{p}{\rho g a} \\ &= i\frac{\sigma^*(x)}{\pi B_O(x)} \psi(y,z;x) e^{-i\nu x} \end{aligned} \tag{13}$$

where

$$\sigma^*(x) = \frac{\sigma(x)\omega_0}{ga}$$

$$\psi(y,z;x) = e^{\nu z} + \Phi(y,z;x)$$

Likewise, we find the linearized wave amplitude in the near field is given by the expression

$$\frac{\zeta(x,y)}{a} = i\,\frac{\sigma^*(x)}{\pi B_o(x)}\;\psi(y,0;x)e^{-i\nu x} \tag{14}$$

The exciting forces and moments are found by integrating the pressure over the body surface. As shown in Beck (1979), the vertical, sectional exciting force can be written as

$$f_3(x)e^{-i\nu x} = \left[i\rho\omega_0\,\frac{\sigma(x)}{\pi B_o(x)}\int_{C(x)} d\ell\,\psi(y,z;x)N_3\right]\,e^{-i\nu x} \tag{15}$$

The total heave exciting force and pitch moment about midship are given by the following integrals along the ship length:

$$F_3 = \int_0^L dx\; f_3(x)\;e^{-i\nu x} \tag{16}$$

$$F_5 = \int_0^L dx\;(L/2-x)\;f_3(x)e^{-i\nu x} \tag{17}$$

where

F_3 = total heave exciting force
F_5 = total pitch exciting moment
L = ship length

The wave induced bending moment is found by twice integrating the vertical exciting force up to the desired station. Setting the shear at the bow equal to zero and integrating by parts once, we arrive at

$$\overline{BM}(x) = -\int_0^x d\xi\,(\xi-x)\;f_3(\xi)e^{-i\nu\xi} \tag{18}$$

In the computer program, the value of x, the station at which the bending moment is evaluated, is specified by the user.

NUMERICAL TECHNIQUE

The two-dimensional, near field problem is solved at the input stations. For any station which is a duplicate of the previous station (i.e. parallel mid-body), the results of the previous station are copied rather than resolving the near-field problem. The near-field problem is solved using the method of Troesch (1979). Because Troesch's method has been documented elsewhere (Troesch (1976a), Troesch (1977)), it will not be discussed here. His computer program has been modified to run in head seas by eliminating the imaginary part of the Green function and setting k=ν. Many of Troesch's subroutines have been used directly and others needed only slight modification. It should be noted that Troesch's program was written in double precision, so that some switching of modes between the variables in the present program and the variables in the Troesch subroutines is necessary. This switching occurs in SUBROUTINE SETUP and SUBROUTINE TWDATA.

For accuracy, the Volterra integral equation is solved at more stations along the ship length than just the input stations. The input stations are located at XAXIS(I). The stations for the integral equation and subsequent calculations are located at XI(I). The increased number of stations is developed in SUBROUTINE INSERT. This subroutine inserts more stations so that their spacing is approximately a cosine distribution along the length. The inserted stations are never closer than ± EPSIL to the input station. At present EPSIL is set equal to .5% of the ship length. Figure 2 shows the two axis system along the length. In the figure, there are 7 input stations and 13 XI-stations. The array ISNUM(I) is used to store the number of the XI-station at each input station. For example, in Figure 2 ISNUM(4)=7. The values of the various quantities

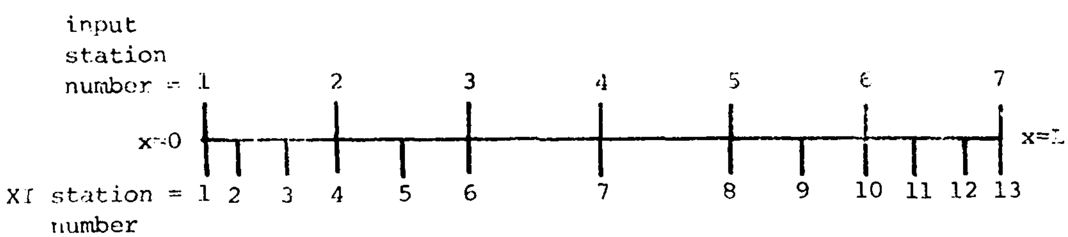

Figure 2. Numbering of the XAXIS(I) and XI(I) axis systems

which are computed in the two-dimensional problem at the input station are determined at the XI-station by cubic-spline interpolation in SUBROUTINE INTRPL.

The Volterra integral equation (equation (3)) is solved at each XI station by a marching process starting at the bow. $\sigma(x)$ is assumed to vary linearly over each segment. The value of $\sigma(x)$ between the ξ_{j+1} and ξ_j station is then given by

$$\sigma(\xi) = \frac{(\xi-\xi_j)}{\Delta_j}\sigma_{j+1} + \frac{(\xi_{j+1}-\xi)}{\Delta_j}\sigma_j \tag{19}$$

where

σ_j = value of $\sigma(x)$ at station number j

σ_{j+1} = value of $\sigma(x)$ at station number j+1

$\Delta_j = \xi_{j+1} - \xi_j$

To develop the marching process we first rewrite the integral equation as

$$\frac{ga}{\omega_0} + \sigma_{j+1}\left[\frac{1}{\pi B_{0_{j+1}}} - C\right] - \alpha\sum_{k=1}^{j}\int_{x_k}^{x_{k+1}} d\xi \frac{\sigma(\xi)}{\sqrt{\xi_{j+1}-\xi}} = 0 \tag{20}$$

where $B_{0_{j+1}}$ equals the value of B_0 at station number j+1. The integral in equation (20) can be evaluated analytically by substituting equation (19) as follows

$$I = \frac{1}{\Delta_k}\int_{x_k}^{x_{k+1}} d\xi \frac{(\xi-\xi_k)\sigma_{k+1} + (\xi_{k+1}-\xi)\sigma_k}{\sqrt{\xi_{j+1}-\xi}}$$

$$= \frac{\sigma_{k+1}}{\Delta_k}\left[G(\xi_k,\xi_{j+1},\xi_{k+1}) - G(\xi_k,\xi_{j+1},\xi_k)\right]$$

$$- \frac{\sigma_k}{\Delta_k}\left[G(\xi_{k+1},\xi_{j+1},\xi_{k+1}) - G(\xi_{k+1},\xi_{j+1},\xi_k)\right] \tag{21}$$

where $G(\alpha,\beta,\gamma) = (2\alpha - \frac{4\beta}{3} - \frac{2\gamma}{3})\sqrt{\beta-\gamma}$ (22)

Multiplying both sides of equation (20) by $2\pi B_{0_{j+1}}$ and substituting equation (22) we arrive at the following expression for σ_{j+1} in terms of all the previous σ_j.

$$\sigma_{j+1}\left[2-2\pi C\, B_{o_{j+1}} - 2\pi B_{o_{j+1}} \frac{\alpha}{\Delta_j}\{G(\xi_j,\xi_{j+1},\xi_{j+1})-G(\xi_j,\xi_{j+1},\xi_j)\}\right]$$

$$= -2\pi B_{o_{j+1}} \frac{ga}{\omega_0} - 2\pi B_{o_{j+1}} \frac{\alpha\sigma_j}{\Delta_j}\{G(\xi_{j+1},\xi_{j+1},\xi_{j+1})-G(\xi_{j+1},\xi_{j+1},\xi_j)\}$$

$$+ 2\pi B_{o_{j+1}}\alpha \sum_{k=1}^{j-1} \frac{1}{\Delta_k}\left[\sigma_{k+1}\{G(\xi_k,\xi_{j+1},\xi_{k+1})-G(\xi_k,\xi_{j+1},\xi_k)\}\right.$$

$$\left. - \sigma_k\{G(\xi_{k+1},\xi_{j+1},\xi_{k+1})-G(\xi_{k+1},\xi_{j+1},\xi_k)\}\right]$$

(23)

In equation (23) the function $G(\alpha,\beta,\gamma)$ is given by equation (22).

To start the marching process, it is assumed that the near-field and three-dimensional source strengths are zero at the bow (i.e. $\sigma_1 = B_{0_1} = 0$). By expanding the integral equation for $\sigma(x)$ around $x=0$ and taking the limit as $x \to 0$, it can be shown that

$$\sigma'(x) = -\pi \frac{ga}{\omega_0} B_o'(x) \quad \text{at } x=0 \tag{15}$$

where the prime denotes differentiation with respect to x. The linear approximation used for $\sigma(x)$ in the numerical scheme leads to exactly this same slope at $x=0$.

At the stern, there are two possible cases. For cruiser sterns, $\sigma(x)$ and $B_0(x)$ are set equal to zero. For transom sterns, the program uses the values of $B_0(L)$ and $\sigma(L)$ as computed for the transom stern section. Thus, $\sigma(x)$ and $B_0(x)$ are not equal to zero at the stern section. The validity of this result should be further investigated, but it seems to give reasonable answers.

The pressure and near-field wave amplitude are computed by equations (13) and (14). In computing these quantities, the quotient $\sigma(x)/\pi B_0(x)$ can not be

computed for points at which $B_0(x)=0$ (i.e. at the bow and for cruiser sterns). The proper limits for the quotient are found by expanding the integral equation for $\sigma(x)$ around x=0 or x=L and taking the proper limits. At the bow, we find

$$\lim_{x\to 0} \frac{\sigma(x)}{\pi B_0(x)} = \frac{ga}{\omega_0}$$

At the stern, the limit need only be taken for cruiser type sterns. In this case, the result is

$$\lim_{x\to L} \frac{\sigma(x)}{\pi B_0(x)} = -\frac{ga}{\omega_0} + \alpha \int_0^L d\xi \, \frac{\sigma(\xi)}{\sqrt{L-\xi}}$$

The exciting forces and midship bending moment are found from the evaluation of equations (15), (16), (17) and (18). These equations all involved integrals of the form

$$I = \int_0^x d\xi \; f(\xi) e^{-i\nu\xi}$$

where $f(\xi)$ is a function which varies smoothly along the length. To evaluate these integrals a simple Filon-Trapezoidal rule is used (see Tuck (1967). This integration is carried out in SUBROUTINE TRAP.

REQUIRED INPUT

The required input is subdivided into two separate parts. The first part contains all the control information such as number of stations, wave frequencies, ship speed, etc. The second part is the offsets which describe the ship hull. The control information is read in on device number 5. The ship offsets are read in on device 7.

Control Information (Read in on device 5)

Card 1: FORMAT (7I5)

- NSTA - number of ship stations to be read in
- ND - number of divisions to be used in developing cosine station spacing used in setting up the XI(I) axis system.
- NVEL - number of velocities at which calculations are to be made (maximum of 8)
- NFREQ - number of wave frequencies at which calculations are to be made (maximum of 16)
- NPRES - number of stations at which pressure information is desired (maximum of NSTA)

NWAVE - control constant for wave amplitude calculation
= 0 no wave amplitude calculation
=1 compute wave amplitude along ship side

NMID - the number of the station at which the midship bending moment is computed.

Card 2: FORMAT (4F10.4)

RHO - water density (slugs/ft^3)

GRAV - acceleration of gravity (ft/sec^2)

XLBP - ship length (ft)

ZETAO - incident wave amplitude (ft)

Card 3: FORMAT (8F10.4)

FROUD(I) - Froude numbers at which calculation are to be made. There should be NVEL values of FROUD(I).

Card 4: FORMAT (8F10.4)

WLXL(I) Wavelength-to-ship-length ratios at which calculations are to be made. NFREQ values of WIXL(I) should be read in.

Card 5: FORMAT (10I5)

NUM(I) - The numbers of the stations at which the pressure distribution is to be completed. There should be NPRES values of NUM(I). NOTE if NPRES=NSTA or NPRES=0 data card 5 must be omitted.

Ship Offsets (Read in on device 7)

There is one complete set of data cards for each ship station. At least one offset (up to a maximum of 25) must be given for each station. Only one offset should be given at the bow; it may be either the point (0,0) or, for a plum bow, the point (0,T) where T is the fore-foot draft. The ship may have a cruiser (one offset point) stern or a transom (many offset points) stern.

Figure 3 is a picture of the means by which the offsets are read in for a given station. For accuracy in the calculations, more points should be entered near the waterline and in areas where the shape changes rapidly (i.e. the turn of the bilge). The points are always entered starting at the negative waterline and reading counterclockwise. Only half the section should be entered. There must be data points at (-B/2,0) and (0,T).

For each station, the following data cards are necessary.

Card 1: FORMAT (I5,F10.4)

NT2(ISTA) = number of offset points on the half section for the station number ISTA.

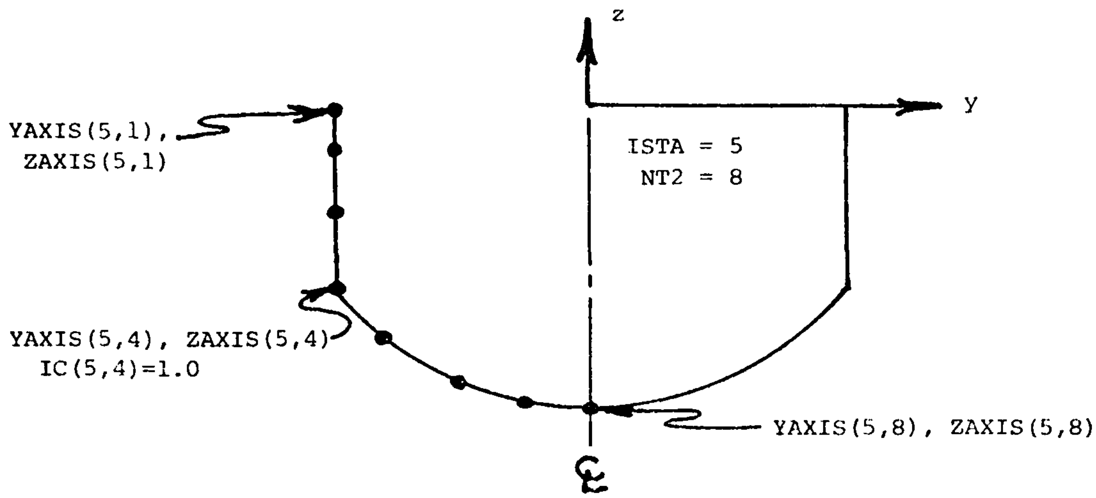

Figure 3. Input offset points around a ship station

XAXIS (ISTA) = x location of the station being read in. x=0 is the bow.

Card 2: FORMAT (2F10.4,I5)

YAXIS(ISTA,I) = y - location of the I^{th} point

ZAXIS(ISTA,I) = z - location of the I^{th} point

IC(ISTA,I) = if IC=1, the I^{th} point is considered a chine point. Normally IC=0.

There should be NT2 data cards number 2 for each station.

Note for each parallel midbody station, NT2(ISTA) is set equal to 999 and data card 2 is omitted. This causes the program to copy the results of the previous station.

Sample input data are shown in Appendix II.

OUTPUT

A sample output listing is shown in Appendix III. The sample is for the input shown in Appendix II.

The first set of output is a listing of the input control constants. In addition, the computed values for the area of the waterplane (AWL) and the full beam (BEAMWL) at the NMID station are listed. These values are used in the nondimensionalization at a later point in the program. It should be noted that the table of offsets read in on device number 7 is not reprinted as output. This was done in order to shorten the output.

The second set of output is a listing of the values of $B_0(x)$ along the ship length. Since $B_0(x)$ does not vary with Froude number, the values are only printed once for each frequency.

The next set is the three-dimensional source strength ($\sigma(x)$) distribution along the ship length. The magnitude of $\sigma(x)$ is printed in the column labeled MAG(SIGMA). The magnitude of the nondimensional sigma and its real and imaginary parts are printed in the subsequent three columns. The nondimensional sigma is defined as

$$\sigma^*(x) = \frac{\sigma(x)\omega_0}{ga}$$

The pressure distribution over each station asked for in the input follows the source distribution listing. If NPRES=0, no pressure distributions are printed. The pressure distribution can only be obtained at the ship stations given in the input. The y,z coordinates and angle up from the keel are printed in the first three columns. The two columns under the heading "2-D POTENTIAL" correspond to the two terms of equation (7). PFRE equals $e^{+\nu z}$ and PHRE gives the values of $\Phi(y,z;x)$. The magnitude and phase angle of the nondimensional pressure are printed in the last two columns. The pressure is computed by equation (13). The phase angles are all relative to a wave node at the bow.

If NWAVE=1, the wave amplitude along the ship length is computed and printed out. The nondimensional wave amplitude is computed by equation (14) and includes both the incident plus diffracted wave. As with the pressure, the phase angle is relative to a wave node at the bow.

The sectional exciting force distribution is print out under the heading EXCITING FORCE DISTRIBUTION. The sectional exciting force ($f_3(x)$) is computed using the expression inside the square brackets of equation (15). Note that the $e^{-i\nu x}$ is not included in the expression for $f_3(x)$. In the print out, the sectional exciting force is nondimensionalized in the following manner

$$F_3(x) = \frac{f_3(x)}{ga\ B(x)/2}$$

where $B(x)/2$ is the LOCAL half beam. The use of the local half beam facilitates comparisons between F3 and purely two-dimensional calculations.

Finally, the total heave force, pitch moment about midship and the bending moment at station number NMID are printed out. The results are computed using

equations (16), (17), and (18). The printed results are nondimensionalized in the following manner:

$$F_3^* = \frac{F_3}{\rho g a L B}$$

$$F_5^* = \frac{F_5}{\rho g a L^2 B}$$

$$\overline{BM}^* = \frac{\overline{BM}}{\rho g a L^2 B}$$

where

L = ship length

B = full beam at the waterline of the NMID station.

$\overline{BM}$ = bending moment at NMID station.

The phase angles are all relative to a wave node at the bow.

LIMITATIONS OF THE PROGRAM

As presently written, the program has several limitations of which the user should be aware. The limitations of the subprograms which calculate the two-dimensional solution are discussed by Troesch (1976b). When the hull section shape is very thin or has areas of high curvature more input points are needed. As the number of input points is increased, the run time of the program will be greatly increased. Furthermore, there are eigen frequencies at which the solution blows up as described by Troesch (1976b). this is a result of the use of an integral-equation technique to solve the two-dimensional problem. It only occurs at high frequencies and for normal ship operating ranges should be no problem. It should be noted that at springing frequencies the effects of the eigen frequencies may become apparent.

The second major limitation has to do with end effects. As with any slender-body theory, the results near the ends are of questionable accuracy. The proper means of handling large bulbous bows and transom sterns is not obvious. At the bow, the program can not handle bulb sections which protrude underwater forward of the fore-perpendicular. The program can handle normal bulb sections which intersect the free surface. Thus, the user must "fair out" the protruding section of the bow. In the stern region, it is important

that the input data, including the last station, be fair in the longitudinal direction. In particular, for cruiser sterns, where the offsets of the last station are (0.0, 0.0), it has been observed that the predicted wave amplitude sometimes shows a marked change from the previous station. This type of result at the stern section should be viewed with caution.

The final limitation deals with the forward speed results. The theoretical analysis (see equation (3)) was carried out for the two speed ranges $\tau=0$ and $\tau>>.25$. The range between these two extremes has not been formally analyzed. The program as presently written has a switch in it at $\tau=.25$. For $\tau<.25$, the zero speed value of C is used, and for $\tau>.25$, C is set equal to the large τ value. This switch can lead to a discontinuity in the results at $\tau=.25$.

It should be pointed out that the theoretical analysis assumes high frequencies (or short wavelengths). The program appears to give reasonable results over the entire frequency range. However, in deriving the expression for the pressure (equation (13)) certain forward speed terms were neglected because of the high frequency assumption. Under certain combinations of forward speed and wave frequency these terms may become important.

LIST OF SUBROUTINES

BESINT - initializes tables of I Bessel functions for use in determining the Green's function for the Helmholtz problem.

BESK - computes the K Bessel function for a given argument and order.

BIK - is a single precision subroutine that evaluates the integral of $K_0(t)$ for t ranging from zero to X.

BK1MOD - computes the $K_1(X)$ Bessel function minus its 1/X singularity for a given argument.

DBS - solves a system of simultaneous equations using back substitution. The L-U decomposition of the coefficient matrix is computed by DLUD.

DEI, NATSEI, FCNMON, ERRTA - are subroutines that evaluate the real exponential integral. They are part of a mathematics package from Argonne National Laboratory.

DEICOM - calculates the complex exponential integral.

DLUD - computes the L-U decomposition of the coefficient matrix by Gaussian elimination with partial pivoting.

DQL4, DQL8, DQL12 - evaluate the integral exp(-X) times some function of X for X ranging from zero to infinity by a four, eight and twelve point Laguerre quadrature formula respectively. These subroutines are from the IBM-SSP listing.

FNT1, FNT2, FNT3 - are all function subroutines that are called by the DQL routines.

G - function used in computing the integral $\int \sigma/\sqrt{x-\xi}\, d\xi$

GRFUN - evaluates the complex Green's function that satisfies the Helmholtz equation in the fluid domain and the linear free surface boundary conditions.

[illegible] - finds the [illegible] parameters needed by the main program. These include the normal, the curvature, and the arc length for the point in question. The parameters are determined after a circular arc is fitted through these points.

INSERT - inserts stations along ship length so that a cosine station spacing results.

INTRPL - interpolates values along ship length using piecewise cubic fit.

NORM - determines the normal to a line drawn between two points.

PRESS - computes the pressure distribution around a ship station.

SAN[illegible] - evaluates the sine and cosine integrals.

S[illegible] - reads ship offsets from device 7 and computes various geometric quantities.

SOURCE - computes the 3-D source strength by solving the Volterra integral equation.

SUM - a complex function which computes the integral $\int_0^x \sigma(\xi)/\sqrt{x-\xi}\, d\xi$

SXY - finds a series that helps define the source potential in GRFUN.

TRAP - computes the integral $\int f(x) e^{-i\nu x}\, dx$

TWDATA - computes the 2-D potential, exciting forces and $E_0(x)$ using results from TWODIM.

TWODIM - computes the solution to the 2-D problem using Frank's method.

WAVE - finds the wave amplitude along the ship length.

REFERENCES

1. Beck, R.F., "Wave Diffraction in Head Seas," to be published.

2. Troesch, A.W., "The Diffraction Potential for A Slender Ship Moving Through Oblique Waves," Report No. 179, Department of Naval Architecture and Marine Engineering, The University of Michigan, Ann Arbor, Michigan, 1976a.

3. Troesch, A.W., "Computer Program that Solves a 2-D Diffraction Problem of Incident Waves on a Symmetric Body and Satisfying a Helmholtz Equation," Report to Naval Sea Systems Command, General Hydromechanics Research Program, Department of Naval Architecture and Marine Engineering, The University of Michigan, Ann Arbor, Michigan, 1976b.

4. Troesch, A.W., "The Diffraction Forces for a Ship Moving in Oblique Seas," Journal of Ship Research, Vol 23, No. 2, pp. 127-139, 1979.

5. Tuck, E.O., "A Simple Filon-Trapezoidal Rule," Mathematics of Computation, Vol. 21, No. 98, 1967.

6. Ursell, F., "The Expansion of Water-Wave Potentials at Great Distances," Proceedings Cambridge Philosophical Society, Vol. 64, pp. 811-826.

7. Ursell, F., "On Head Seas Travelling Along a Horizontal Cylinder," Journal of The Institute of Mathematics and its Application, Vol. 4, 1968b, pp. 414-427.

APPENDIX I

Program Listing of Diffracted Forces Program

```
                                                                        ISN
CC
C     PROGRAM TO COMPUTE THE EXCITING FORCES ON A SHIP IN HEAD SEAS
C
      COMMON RHO,GRAV,XLBP,HXLBP,XAXIS(21),YAXIS(21,25),                  1
     1ZAXIS(21,25),BEAM(21),
     2XI(50),XDEL(50),FROUD(8),U,XLAMDA(16),WAVEN(16),OMEGA(16),
     3AA(16),OMEGAE,TAU,TAUS,NSTA,NDIV,NDIV1
      COMMON /TWOD/ PHRE(21,50),PERE(21,50),DO(21),FKVRE(21),DPVRE(21)    2
      COMMON /SHIPD/ X1(25),Y1(25),BFMAX(21),AN1(21,50),                  3
     1AN2(21,50),ACURV(21,50),ARC1(21,25),ARC2(21,25),IC(21,25),
     2NT(21),NT2(21),NT5(21)
      REAL*8 X1,Y1,BEMAX,AN1,AN2,ACURV,ARC1,ARC2                          4
      COMMON /SIG/ SIGMA(50),SIGND(50),SIGNDO(21),ALPHA,BX(50),           5
     1SIGMAM(50),SIGNDM(50)
      COMPLEX IM,SIGMA,SIGND,SIGNDO,F3,DUM,F3TOT,FSTOT,ALPHA,             6
     +        BEND,EXCIT
      REAL*8 COEF                                                         7
      DIMENSION WLXL(16),SIGNDR(50),SIGNDI(50),BEAMX(50),F3M(50),F3(50),  8
     1F3P(50),DUM(50),NUM(21),NCHECK(21),ISNUM(21)
      DIMENSION FKVREX(50),DPVREX(50),SIGOR(21),SIGOI(21)                 9
C
C...  READ INPUT CONTROL DATA, CONSTANTS, FROUDE NUMBERS AND
C...  LAMBDA/LBP RATIOS.
C
      WRITE(6,2000)                                                      10
      READ(5,1000) NSTA,ND,NVEL,NFREQ,NPRES,NWAVE,NMID                   11
         IF(NMID .EQ. 0)NMID=(NSTA+1)/2                                  12
      WRITE(6,1001)NSTA,ND,NVEL,NFREQ,NPRES,NWAVE,NMID                   13
      READ(5,1010) RHO,GRAV,XLBP,ZETAO                                   14
      WRITE(6,1011)RHO,GRAV,XLBP,ZETAO                                   15
      READ(5,1010) (FROUD(I),I=1,NVEL)                                   16
      WRITE(6,1012) (FROUD(I),I=1,NVEL)                                  17
      READ(5,1010) (WLXL(I),I=1,NFREQ)                                   18
      WRITE(6,1013) (WLXL(I),I=1,NFREQ)                                  19
      IF(NPRES.EQ.0) GO TO 16                                            20
      IF(NPRES.NE.NSTA) GO TO 8                                          21
    5 DO 7 I=1,NSTA                                                      22
    7 NUM(I)=I                                                           23
      GO TO 6                                                            24
C
C...  READ STATIONS AT WHICH PRESSURE IS DESIRED.
    8 READ(5,1015) (NUM(I),I=1,NPRES)                                    25
    6 WRITE(6,1016)(NUM(I),I=1,NPRES)                                    26
C     SET NCHECK=1 AT STATIONS WHERE PRESSURE IS DESIRED
      DO 10 I=1,NSTA                                                     27
   10 NCHECK(I)=0                                                        28
      DO 15 I=1,NPRES                                                    29
   15 NCHECK(NUM(I))=1                                                   30
   16 NSTAM1=NSTA-1                                                      31
      HXLBP=.5*XLBP                                                      32
      IM=CMPLX(0.0,1.0)                                                  33
C
C...  SETUP READS THE SHIP OFFSETS AND COMPUTES GEOMETRIC PROPERTIES
C...  FOR USE IN 2-D SOLUTION.
      CALL SETUP                                                         34
C
C...  INSERT ADDS MORE STATIONS ALONG THE SHIP LENGTH AS NEEDED TO
C...  APPROXIMATE A COSINE DISTRIBUTION.
```

```
            CALL INSERT(NSTA,ND,XAXIS,XLBP,NDIV1,XI,ISNUM)
            NDIV=NDIV1-1
            DO 20 I=1,NDIV
            XDEL(I)=XI(I+1)-XI(I)
       20   CONTINUE
      C
      C...  INTERPOLATE BEAM AT XI STATIONS.
            CALL INTRPL(6,NSTA,XAXIS,BEAM,NDIV1,XI,BEAMX)
      C
      C...  COMPUTE WATERPLANE AREA AND BEAM.
            AWL=0.0
            DO 22 I=1,NDIV
       22   AWL=AWL+.5*XDEL(I)*(BEAMX(I)+BEAMX(I+1))
            AWL=2.*AWL
            BWL=BEAM(NMID)*2.
            WRITE(6,2002) AWL,BWL
      C
      C...  START LOOP ON EACH FREQUENCY
            DO 999 IO=1,NFREQ
      C
      C...  COMPUTE WAVE CONSTANTS.
            XLAMDA(IO)=WLXL(IO)*XLPP
            WAVEN(IO)=6.283185/XLAMDA(IO)
            OMEGA(IO)=SQRT(GRAV*WAVEN(IO))
            AA(IO)=ZETAO*GRAV/OMEGA(IO)
            COEF=DELE(WAVEN(IO))
      C
      C      SET VALUES AT END OF SHIP = 0.0
      C...  FOR TRANSOM STERN, ACTUAL VALUES ARE USED.
            DO 25 ISTA=1,NSTA
               IF(NT2(ISTA) .GT. 1)GO TO 21
                  DO(ISTA)=0.0
                  DFVRE(ISTA)=0.0
                  FKVRE(ISTA)=0.0
                  GO TO 25
        21     CONTINUE
      C
      C...  IF PARALLEL MIDDLE BODY, DUPLICATE RESULTS OF 2-D PROBLEM.
            IF(NT2(ISTA).EQ.999) GO TO 23
      C
      C...  TWODIM SOLVES THE 2-D PROBLEM FOR HELMHOLTZ'S EQUATION.
            CALL TWODIM(COEF,YAXIS,ZAXIS,ISTA)
            GO TO 25
      C
      C      EQUATE VARIABLES OF PARALLEL MIDBODY
      C
       23   NT2(ISTA)=NT2(ISTA-1)
            NNT=NT(ISTA)
            DO(ISTA)=DO(ISTA-1)
            FKVRE(ISTA)=FKVRE(ISTA-1)
            DFVRE(ISTA)=DFVRE(ISTA-1)
            DO 24 I=1,NNT
            PHPR(ISTA,I)=PHPR(ISTA-1,I)
            PFRE(ISTA,I)=PFRE(ISTA-1,I)
      24    CONTINUE
      25    CONTINUE
      C
      C...  INTERPOLATE DO AT XI STATIONS.
```

```
      CALL INTSPL(6,NSTA,XAXIS,BC,NDIV1,XI,BX)                        73
C
C...  WRITE HEADINGS FOR EACH FREQUENCY.
      WRITE(6,2001) WLXL(IO),WAVEN(IO),OMEGA(IO)                      74
      WRITE(6,2003)                                                   75
      WRITE(6,2005)                                                   76
C
C...  WRITE BO
      WRITE(6,2010)  (XI(I),BX(I),I=1,NDIV1)                          77
C
C...  START FROUDE NUMBER LOOP.
      DO 999 IFR=1,NVEL                                               78
      U=FROUD(IFR)*SQRT(XLBP*GRAV)                                    79
      OMEGAE=OMEGA(IO)+WAVEN(IO)*U                                    80
      TAU=OMEGAE*U/GRAV                                               81
      TAUS=OMEGA(IO)*U/GRAV                                           82
C
C...  SOURCE COMPUTES THE 3-D SOURCE STRENGTH.
      CALL SOURCE(IO,IFR)                                             83
      WRITE(6,2015)  FROUD(IFR),U,OMEGAE,TAU,TAUS                     84
      WRITE(6,2016)                                                   85
      WRITE(6,2020)                                                   86
C...  WRITE THE 3-D SOURCE STRENGTH.
      WRITE(6,2025)  (XI(I),SIGMAM(I),SIGMDM(I),SIGND(I),I=1,NDIV1)   87
      IF(NPRES.EQ.0) GO TO 41                                         88
      DO 40 ISTA=1,NSTA                                               89
C...  FIND PRESSURE AT DESIRED STATIONS.
      IF(NCHECK(ISTA).NE.1) GO TO 40                                  90
C...  PRESUR COMPUTES THE PRESSURE DISTRIBUTION OVER THE SHIP STATION.
      CALL PRESUR(IO,ISTA,ISNUM(ISTA))                                91
   40 CONTINUE                                                        92
C...  WAVE COMPUTES WAVE AMPLITUDE ALONG THE SHIP LENGTH.
   41 IF(NWAVE.EQ.0) GO TO 42                                         93
      CALL WAVE(IO,ISNUM)                                             94
C...  INTERPOLATE 2-D EXCITING FORCES AT XI STATIONS.
   42 CALL INTSPL(6,NSTA,XAXIS,DPVRE,NDIV1,XI,DPVREX)                 95
      CALL INTSPL(6,NSTA (AXIS,FKVRE,NDIV1,XI,FKVREX)                 96
C...  COMPUTE EXCITING FORCE AT EACH STATION.
      DO 45 I=1,NDIV1                                                 97
      F3(I)=IM*SIGND(I)*.31831*(FKVREX(I)+DPVREX(I))                  98
C...  ONLY DIVIDE BY BO IF IT IS NOT EQUAL TO 0
      IF(I.NE.1.AND.I.NE.NDIV1) F3(I)=F3(I)/BX(I)                     99
      IF(I.EQ.NDIV1.AND.NT2(NSTA).GT.1) F3(I)=F3(I)/BX(I)             100
C...  ELIMINATE THE LOCAL HALF-BEAM, WHICH IS USED TO
C...  NONDIMENSIONALIZE F3(I).
      DUM(I)=F3(I)*BEAMX(I)                                           101
C...  COMPUTE MAGNITUDE AND PHASE OF SECTIONAL EXCITING FORCE.
      F3M(I)=CABS(F3(I))                                              102
      IF(AIMAG(F3(I)).NE.0.0.OR.REAL(F3(I)).NE.0.0) GO TO 46          103
      F3P(I)=0.0                                                      104
      GO TO 45                                                        105
   46 F3P(I)=ATAN2(AIMAG(F3(I)),REAL(F3(I)))*57.29578                 106
   45 CONTINUE                                                        107
C...  WRITE SECTIONAL EXCITING FORCES.
      WRITE(6,2050)                                                   108
      WRITE(6,2055) (XI(I),F3(I),F3M(I),F3P(I),I=1,NDIV1)             109
C...  TRAP COMPUTES INTEGRAL (F*EXP(-I*WAVEN*XI)*BX) TO FIND
C...  TOTAL EXCITING FORCE.
```

```
                                                                          ISN
      CALL TRAP(XI,DUM,NDIV1,WAVEN(IO),XDEL,F3TOT)                        110
C...  COMPUTE MAGNITUDE AND PHASE OF TOTAL HEAVE FORCE.
      F3TOT=F3TOT/(XLBP*BWL)                                              111
      F3TOTM=CABS(F3TOT)                                                  112
      IF(AIMAG(F3TOT).NE.0.0.OR.REAL(F3TOT).NE.0.0) GO TO 48              113
      F3TOTP=0.0                                                          114
      GO TO 49                                                            115
   48 F3TOTP=ATAN2(AIMAG(F3TOT),REAL(F3TOT))*57.29578                     116
C...  COMPUTE PITCH EXCITING MOMENTS.
   49 DO 50 I=1,NDIV1                                                    117
   50 DUM(I)=DUM(I)*(HXLBP-XI(I))                                         118
      CALL TRAP(XI,DUM,NDIV1,WAVEN(IO),XDEL,F5TOT)                        119
      F5TOT=F5TOT/(XLBP*XLBP*BWL)                                         120
      F5TOTM=CABS(F5TOT)                                                  121
      IF(AIMAG(F5TOT).NE.0.0.OR.REAL(F5TOT).NE.0.0) GO TO 51              122
      F5TOTP=0.0                                                          123
      GO TO 52                                                            124
   51 F5TOTP=ATAN2(AIMAG(F5TOT),REAL(F5TOT))*57.29578                     125
   52 CONTINUE                                                            126
C...  WRITE TOTAL HEAVE AND PITCH EXCITING FORCE AND MOMENT.
      IMID=ISNUM(NMID)                                                    127
      WRITE(6,2060)XI(IMID)                                               128
      WRITE(6,2065) F3TOT,F3TOTM,F3TOTP                                   129
      WRITE(6,2070) F5TOT,F5TOTM,F5TOTP                                   130
C
C...  COMPUTE MIDSHIP BENDING MOMENT DUE TO EXCITING FORCES.
      DO 53 I=1,IMID                                                      131
        DUM(I)=F3(I)*BEAMX(I)*(XI(I)-XI(IMID))                            132
   53 CONTINUE                                                            133
C
      CALL TRAP(XI,DUM,IMID,WAVEN(IO),XDEL,BEND)                          134
C
      BEND=-1.*BEND/(XLBP*XLBP*BWL)                                       135
      BENDM=CABS(BEND)                                                    136
      IF(AIMAG(BEND) .NE. 0.0 .OR. REAL(BEND) .NE. 0.0)                   137
     *     BENDP=ATAN2(AIMAG(BEND),REAL(BEND))*57.29578
      IF(AIMAG(BEND) .EQ. 0.0 .AND. REAL(BEND) .EQ. 0.0)BENDP=0.0         138
C...  WRITE MIDSHIP BENDING MOMENT.
      WRITE(6,2075)BEND,BENDM,BENDP                                       139
C
  999 CONTINUE                                                            140
      STOP                                                                141
 1000 FORMAT(7I5)                                                         142
 1001 FORMAT(' NSTA=',I5,'     ND=',I5,'   NVEL=',I5/                     143
     *        'NFREQ=',I5,'  NOBES=',I5,'  NWAVE=',I5,'  NMID=',I5)
 1010 FORMAT(8F10.4)                                                      144
 1011 FORMAT('    RHO=',F10.4,'  GRAV=',F10.4,'  XLBP=',F10.4/            145
     *       '  DELTAC=',F10.4)
 1012 FORMAT('FROUDE NUMBERS'/(8F10.4))                                   146
 1013 FORMAT('WAVE LENGTHS (LAMBDA/LBP)'/(8F10.4))                        147
 1015 FORMAT(16I5)                                                        148
 1016 FORMAT('STATIONS AT WHICH PRESSURE IS DESIRED'/(10I5))              149
 2000 FORMAT(1H1,'    INPUT DATA')                                        150
 2001 FORMAT(1H1,'   WLXL=',F10.4,' WAVEN=',F10.4,' OMEGA=',F10.4)        151
 2002 FORMAT('  BWL=',E15.5,'  BEAM WL=',E15.5)                           152
 2003 FORMAT(1H0,'TWO DIMENSIONAL SOURCE STRENGTH DISTRIBUTION')          153
 2005 FORMAT(1H0,'   XI',15X,'BO')                                        154
 2010 FORMAT(F10.4,E15.5)                                                 155
```

```
                                                                  ISN
      2015 FORMAT(1H1,'   PROUDE NUM=',F10.4,'  VELOCITY=',F10.4,/     156
          1'   OMEGAE=',F10.4,'  TAU=',F10.4,'  TAU*=',F10.4)
      2016 FORMAT(1H0,'THREE DIMENSIONAL SOURCE STRENGTH DISTRIBUTION')  157
      2020 FORMAT(1H0,'    XI',7X,'MAG(SIGMA)',4X,'MAG(SIGMAND)',4X,      158
          1'RE(SIGMAND)',4X,'IMAG(SIGMAND)')
      2025 FORMAT(F10.4,4E15.5)                                          159
      2050 FORMAT(1H0,'   EXCITING FORCE DISTRIBUTION'/5X,'X',8X,         160
          1'REAL(F3)',7X,'IMAG(F3)',7X,'MAG(F3)',7X,'ARG(F3)')
      2055 FORMAT(F10.4,3E15.5,F10.4)                                    161
      2060 FORMAT(1H0,'   TOTAL EXCITING FORCES AND B.M. AT X=',F10.4/    162
          113X,'REAL',13X,'IMAG',10X,'MAG',8X,'PHASE')
      2065 FORMAT('  HEAVE=',3E  .5,F10.4)                               163
      2070 FORMAT('  PITCH=',3E15.5,F10.4)                               164
      2075 FORMAT('   B.M.=',3E15.5,F10.4)                               165
      END                                                                166
```

```
      SUBROUTINE TRAP(XI,F,N,WAVEN,XDEL,SUM)
C
C     +----------------------------------------------+
C     |                                              |
C     |     THIS ROUTINE COMPUTES THE INTEGRAL       |
C     |  (F(X)*EXP(-I*WAVEN*XI)*DX)  USING TRAPEZOIDAL |
C     |  RULE AND FILON TYPE APPROACH.               |
C     |                                              |
C     +----------------------------------------------+
C
C
      REAL XI(1),XDEL(1),WAVEN
      COMPLEX F(1),SUM,IM,E,W1,W2
C
      IM=(0.,1.)
      SUM=(0.,0.)
C
      NITR=N-1
      DO 1000 I=1,NITR
C
         IF(WAVEN*XDEL(I) .LT. .05)GO TO 500
         E=CEXP(IM*WAVEN*XDEL(I))
         W1=(1.0-IM*WAVEN*XDEL(I))*E-1.0
         W2=(1.0+IM*WAVEN*XDEL(I))-E
C
         SUM=SUM+(F(I)*W1+F(I+1)*W2)/(WAVEN*WAVEN*XDEL(I))
     +        *CEXP(-IM*WAVEN*XI(I+1))
         GO TO 1000
  500    CONTINUE
            SUM=SUM+XDEL(I)*CEXP(-IM*WAVEN*XI(I+1))
     +           *(.50*(F(I)+F(I+1))+IM*WAVEN*XDEL(I)
     +           *(F(I)/3.+F(I+1)/6.))
C
 1000 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE PRESUR(IO,ISTA,ISN)
C
C...  THIS ROUTINE COMPUTES THE PRESSURE DISTRIBUTION ON A SHIP SECTION.
C
      COMMON RHO,GRAV,XLBP,HXLBP,XAXIS(21),YAXIS(21,25),
     1ZAXIS(21,25),BEAM(21),
     2XI(50),XDEL(50),FROUD(8),U,XLAMDA(16),WAVEN(16),OMEGA(16),
     3AA(16),OMEGAE,TAU,TAUS,NSTA,NDIV,NDIV1
      COMMON /TWOD/ PHRE(21,50),PFRE(21,50),BO(21),FKVRE(21),DPVRE(21)
      COMMON /SIG/ SIGMA(50),SIGND(50),SIGNDO(21),ALPHA,BX(50),
     1SIGMAN(50),SIGNDN(50)
      COMMON /SHIPD/ X1(25),Y1(25),BEMAX(21),AN1(21,50),
     1AN2(21,50),ACURV(21,50),ARC1(21,25),ARC2(21,25),IC(21,25),
     2NT(21),NT2(21),NT5(21)
      REAL*8 X1,Y1,BEMAX,AN1,AN2,ACURV,ARC1,ARC2
      COMPLEX IM,SIGMA,SIGND,SIGNDO,CONST,PND,ALPHA,SUM
      DIMENSION PND(25),PMAG(25),PANG(25),THETA(25)
      IM=(0.0,1.0)
      NNT=NT(ISTA)
      NNT2=NT2(ISTA)
C...  COMPUTE CONSTANT FOR FIRST AND LAST STATION.
      IF(ISTA.EQ.1) CONST=-IM
      IF(ISTA.NE.NSTA.OR.NNT2.GT.1) GO TO 3
      CONST=1.-ALPHA*(SUM(NDIV-1)-SIGMA(NDIV)*(G(XLBP,XLBP,XLBP)-
     1G(XLBP,XLBP,XI(NDIV)))/XDEL(NDIV))/AA(IO)
      CONST=-IM*CEXP(-IM*WAVEN(IO)*XLBP)*CONST
    3 IF(ISTA .NE. 1 .AND. ISTA .NE. NSTA)GO TO 8
      IF(ISTA .EQ. NSTA .AND. NNT2 .GT. 1)GO TO 8
      PFRE(ISTA,1)=EXP(WAVEN(IO)*ZAXIS(ISTA,1))
      PHRE(ISTA,1)=0.0
      GO TO 9
    8 CONST=IM*SIGND(ISN)*CEXP(-IM*WAVEN(IO)*XAXIS(ISTA))/
     1(3.14159*BO(ISTA))
    9 DO 10 I=1,NNT2
C...  COMPUTE ANGLE UP FROM THE KEEL.
      IF(ZAXIS(ISTA,I) .EQ. 0. .AND. YAXIS(ISTA,I) .EQ. 0.)THETA(I)=0.
      IF(ZAXIS(ISTA,I) .NE. 0. .OR.  YAXIS(ISTA,I) .NE. 0.)
     *THETA(I)=ATAN2(ABS(YAXIS(ISTA,I)),ABS(ZAXIS(ISTA,I)))*57.29578
      PND(I)=CONST*(PFRE(ISTA,I)+PHRE(ISTA,I))
C...  COMPUTE THE MAGNITUDE AND PHASE OF THE PRESSURE.
      PMAG(I)=CABS(PND(I))
      PANG(I)=ATAN2(AIMAG(PND(I)),REAL(PND(I)))*57.29578
   10 CONTINUE
C...  WRITE OUT THE PRESSURE DISTRIBUTION.
      WRITE(6,2000) ISTA,XAXIS(ISTA)
      WRITE(6,2005)
      WRITE(6,2010)  (YAXIS(ISTA,I),ZAXIS(ISTA,I),THETA(I),
     1PFRE(ISTA,I),PHRE(ISTA,I),PMAG(I),PANG(I),I=1,NNT2)
 2000 FORMAT(1H0,'  PRESSURE DISTRIBUTION FOR STATION',I5,'  X=',
     1F10.4)
 2005 FORMAT(35X,'2-D POTENTIAL',6X,'3-D PRESSURE',/,
     15X,'Y',8X,'Z',9X,'THETA',4X,'  PFRE  ',2X,
     2'  PHRE  ',1X,'MAG(PRES)',1X,'ARG(PRES)')
 2010 FORMAT(7F10.4)
      RETURN
      END
```

```
      SUBROUTINE INSERT(NSTA,ND,XAXIS,XLBP,NDIV1,XI,ISNUM)
C
C     *-----------------------------------------------------------*
C     |                                                           |
C     |     THIS ROUTINE SETS UP THE XI AXIS SYSTEM.  IT          |
C     |  STARTS WITH THE SHIP STATIONS AND INSERTS STATIONS       |
C     |  TO APPROXIMATE A COSINE SPACING.  ISNUM(ISTA) IS         |
C     |  THE NUMBER OF THE XI STATION AT THE ISTA INPUT           |
C     |  STATION.                                                 |
C     |                                                           |
C     *-----------------------------------------------------------*
C
      DIMENSION XAXIS(1),XI(1),ISNUM(1)
      EPSIL=.005*XLBP
      ISNUM(1)=1
      ISTA=2
      I=2
      J=1
      XI(1)=0.0
    2 XIDUM=.5*XLBP*(1.-COS(3.14159*FLOAT(J)/FLOAT(ND)))
C...  CHECK TO MAKE SURE INSERTED STATION IS NOT CLOSER THAN
C...  EPSIL TO A SHIP STATION.
      IF(J.GT.ND) XIDUM=XLBP+2.*EPSIL
    3 IF(ABS(XIDUM-XAXIS(ISTA)).GT.EPSIL) GO TO 4
      J=J+1
      GO TO 2
    4 IF(XIDUM.GT.XAXIS(ISTA)) GO TO 5
      XI(I)=XIDUM
      IF(XI(I).GE.XLBP) GO TO 40
      I=I+1
      J=J+1
      GO TO 2
    5 XI(I)=XAXIS(ISTA)
      IF(XI(I).GE.XLBP) GO TO 40
      ISNUM(ISTA)=I
      I=I+1
      ISTA=ISTA+1
      GO TO 3
   40 ISNUM(ISTA)=I
      NDIV1=I
      IF(NDIV1.LT.50) GO TO 45
      WRITE(6,1000)
 1000 FORMAT(' NDIV1 TOO LARGE, STOP IN ROUTINE INSERT')
      STOP
   45 CONTINUE
      RETURN
      END
```

| DATE:08-29-79,11:33 OWNER:K3AM FILE:SETUP.S |

ISN

```
      SUBROUTINE SETUP
C
C...  THIS ROUTINE READS AND SETS-UP THE SHIP OFFSETS.
C
      REAL*8 X1,Y1,BEMAX,AN1,AN2,ACURV,ARC1,ARC2
      REAL*8 X4,Y4
      COMMON RHO,GRAV,XLBP,HXLBP,XAXIS(21),YAXIS(21,25),
     1ZAXIS(21,25),BEAM(21),
     2XI(50),XDEL(50),FROUD(8),U,XLAMDA(16),WAVEN(16),OMEGA(16),
     3AA(16),OMEGAE,TAU,TAUS,NSTA,NDIV,NDIV1
      COMMON /SHIPE/ X1(25),Y1(25),BEMAX(21),AN1(21,50),
     1AN2(21,50),ACURV(21,50),ARC1(21,25),ARC2(21,25),IC(21,25),
     2NT(21),NT2(21),NT5(21)
      DO 99 ISTA=1,NSTA
      READ(7,1000) NT2(ISTA),XAXIS(ISTA)
      IF(NT2(ISTA).EQ.999) GO TO 90
      NNT2=NT2(ISTA)
      NT(ISTA)=2*NT2(ISTA)-1
      DO 2 I=1,25
    2 IC(ISTA,I)=0
      READ(7,1010) (YAXIS(ISTA,I),ZAXIS(ISTA,I),IC(ISTA,I),I=1,NNT2)
C
C      SWITCH TO TROESCH AXIS SYSTEM
C
      DO 10 I=1,NNT2
      X1(I)=YAXIS(ISTA,I)
      Y1(I)=-ZAXIS(ISTA,I)
   10 CONTINUE
      BEMAX(ISTA)=DABS(X1(1))
      BEAM(ISTA)=BEMAX(ISTA)
      IF(NNT2 .EQ. 1) GO TO 99
      IF(BEMAX(ISTA).GE.1.D-8) GO TO 3
      WRITE(6,2000) ISTA
 2000 FORMAT(1H1,'***ERROR-THE WATERLINE BEAM IS ZERO FOR STATION'
     1,I5)
      STOP
    3 CONTINUE
C
C        THE HULL PARAMETERS: NARMAL, ARCLENGTH, AND CURVATURE
C        ARE FOUND FOR POINTS 2-NT2. HULLP TAKES THREE POINTS AND
C        RETURNS THESE PARAMETERS FOR THE MIDDLE ONE
C
      DO 9 J=2,NNT2
      IF(J.EQ.NT2(ISTA)) GO TO 7
      X4=X1(J+1)
      Y4=Y1(J+1)
      GO TO 8
    7 X4=-X1(J-1)
      Y4=Y1(J-1)
    8 CALL HULLP(X1(J-1),Y1(J-1),X1(J),Y1(J)
     1,X4,Y4,IC(ISTA,J),AN1(ISTA,J),AN2(ISTA,J),ACURV(ISTA,J),
     2ARC1(ISTA,J),ARC2(ISTA,J))
      IF(J.EQ.NT2(ISTA)) GO TO 9
      AN1(ISTA,2*NT2(ISTA)-J)=-AN1(ISTA,J)
      AN2(ISTA,2*NT2(ISTA)-J)=AN2(ISTA,J)
    9 CONTINUE
C
C        THE HULL PARAMETERS ARE FOUND FOR THE FIRST POINT
```

```
C
      IC(ISTA,1)=1                                                    38
      X4=X1(2)                                                        39
      Y4=-Y1(2)                                                       40
      CALL HULLF(X4,Y4,X1(1),Y1(1),X1(2),Y1(2),                       41
     1IC(ISTA,1),AN1(ISTA,1),AN2(ISTA,1),ACURV(ISTA,1),ARC1(ISTA,1),
     2ARC2(ISTA,1))
      ARC1(ISTA,1)=ARC1(ISTA,1)*.5                                    42
      ARC2(ISTA,1)=ARC1(ISTA,1)                                       43
      AN1(ISTA,1)=-1.D0                                               44
      AN2(ISTA,1)=0.0                                                 45
      AN1(ISTA,NI(ISTA))=1.D0                                         46
      AN2(ISTA,NI(ISTA))=0.0                                          47
      NI5(ISTA)=NI2(ISTA)-1                                           48
      GO TO 99                                                        49
C     EQUATE VARIABLES FOR PARALLEL MIDBODY
   90 NNT2=(NI(ISTA-1)+1)/2                                           50
      NI(ISTA)=NI(ISTA-1)                                             51
      NNT=NI(ISTA)                                                    52
      NI5(ISTA)=NI5(ISTA-1)                                           53
      BEAM(ISTA)=BEAM(ISTA-1)                                         54
      BEMAX(ISTA)=BEMAX(ISTA-1)                                       55
      DO 93 I=1,NNT2                                                  56
      YAXIS(ISTA,I)=YAXIS(ISTA-1,I)                                   57
      ZAXIS(ISTA,I)=ZAXIS(ISTA-1,I)                                   58
   93 CONTINUE                                                        59
      DO 94 I=1,NNT                                                   60
      AN1(ISTA,I)=AN1(ISTA-1,I)                                       61
      AN2(ISTA,I)=AN2(ISTA-1,I)                                       62
      ACURV(ISTA,I)=ACURV(ISTA-1,I)                                   63
      ARC1(ISTA,I)=ARC1(ISTA-1,I)                                     64
      ARC2(ISTA,I)=ARC2(ISTA-1,I)                                     65
   94 IC(ISTA,I)=IC(ISTA-1,I)                                         66
   99 CONTINUE                                                        67
 1000 FORMAT(I5,F10.4)                                                68
 1010 FORMAT(F10.4,I5)                                                69
      RETURN                                                          70
      END                                                             71
```

```
C
C**********************************
C
      SUBROUTINE NORM(X1,Y1,X2,Y2,AN1,AN2)
C
C         GIVEN TWO POINTS, NORM CONSTRUCTS A LINE AND THE NORMAL TO IT. THE
C         NORMAL COMPONENTS ARE AN1 AND AN2.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      R=DSQRT((X2-X1)*(X2-X1)+(Y2-Y1)*(Y2-Y1))
      IF(R .LE. 1.D-7) GO TO 10
      AN1=(Y1-Y2)/R
      AN2=(X2-X1)/R
      RETURN
   10 WRITE(6,20) X1,Y1
      STOP 1
   20 FORMAT(///' ***ERROR***'//5X,'DUPLICATE HULL POINTS AT',
     1 2(F10.4))
      END
```

```
      C
      C**********************************
      C
            SUBROUTINE HULLP(X1,Y1,X2,Y2,X3,Y3,ICHINE,N1,N2,CURV,ARC1,ARC2)        1
      C
      C          HULLP TAKES THREE POINTS, PT1, PT2, PT3, AND RETURNS THE NORMAL,
      C          N1, N2, THE CURVATURE, CURV, AND HALF THE ARC LENGTH TO EITHER SIDE
      C          OF THE MIDDLE POINT, ARC1, ARC2.
      C
            IMPLICIT REAL*8 (A-H,O-Z)                                              2
            REAL*8 N1,N2,N1P,N2P                                                   3
      C
      C          IF ICHINE .GT. 0, THE MIDDLE POINT, PT2, IS DESIGNATED AS A CHINE.
      C
            IF(ICHINE .EQ. 0) GO TO 10                                             4
      C
      C          TWO   LINES    ARE DRAWN BETWEEN PT1,PT2 AND PT2,PT3. THE NORMAL AT
      C          PT2 IS THE AVERAGE OF THE NORMALS TO THE TWO LINES.
      C
            CURV=0.D0                                                              5
            CALL NORM(X1,Y1,X2,Y2,AN11,AN12)                                       6
            CALL NORM(X2,Y2,X3,Y3,AN21,AN22)                                       7
            RN=DSQRT(2.D0*(1.D0+AN11*AN21+AN12*AN22))                              8
            N1=(AN11+AN21)/RN                                                      9
            N2=(AN12+AN22)/RN                                                     10
            ARC1=DSQRT((X1-X2)*(X1-X2)+(Y1-Y2)*(Y1-Y2))*0.5D0                     11
            ARC2=DSQRT((X2-X3)*(X2-X3)+(Y2-Y3)*(Y2-Y3))*0.5D0                     12
            RETURN                                                                13
         10 A=X1*(Y2-Y3)-Y1*(X2-X3)+(X2*Y3-X3*Y2)                                 14
      C
      C          CHECK TO SEE IF THE THREE POINTS LIE ON A STRAIGHT LINE, BEFORE
      C          A CIRCLE IS FITTED. THE EQUATION FOR THE CIRCLE COMES FROM THOMAS,
      C          CALCULUS AND ANALYTIC GEOMETRY, PAGE 463.
      C
            IF(DABS(A) .GE. 1.D-5) GO TO 20                                       15
            CALL NORM(X1,Y1,X3,Y3,N1,N2)                                          16
            CURV=0.D0                                                             17
            ARC1=DSQRT((X1-X2)*(X1-X2)+(Y1-Y2)*(Y1-Y2))*0.5D0                     18
            ARC2=DSQRT((X2-X3)*(X2-X3)+(Y2-Y3)*(Y2-Y3))*0.5D0                     19
            RETURN                                                                20
         20 SQ1=X1*X1+Y1*Y1                                                       21
            SQ2=X2*X2+Y2*Y2                                                       22
            SQ3=X3*X3+Y3*Y3                                                       23
            D=-(SQ1*(Y2-Y3)-Y1*(SQ2-SQ3)+(Y3*SQ2-Y2*SQ3))                         24
            E=SQ1*(X2-X3)-X1*(SQ2-SQ3)+(SQ2*X3-SQ3*X2)                            25
            F=-(SQ1*(X2*Y3-X3*Y2)-X1*(SQ2*Y3-SQ3*Y2)+Y1*(SQ2*X3-SQ3*X2))          26
      C
      C          THE CENTER OF THE CIRCLE IS GIVEN AT (XO,YO) AND THE RADIUS IS R.
      C
            XO=-D/(2.D0*A)                                                        27
            YO=-E/(2.D0*A)                                                        28
            R=DSQRT(XO*XO+YO*YO-F/A)                                              29
            CALL NORM(X1,Y1,X2,Y2,N1P,N2P)                                        30
            SGN=1.D0                                                              31
      C
      C          THE DOT PRODUCT BETWEEN THE VECTOR FROM THE CENTER OF THE CIRCLE
      C          TO THE MID POINT OF A LINE BETWEEN PT1 AND PT2, AND THE OUTWARD
      C          NORMAL AT THAT MID POINT TELLS WHETHER THE CURVE IS CONCAVE OR
```

```
C         CONVEX, AND HENCE GIVES THE SIGN ON THE CURVATURE AND NORMAL
C         COMPONENTS.
C
      DOTP=((X1+X2)/2.D0-X0)*N1P+((Y1+Y2)/2.D0-Y0)*N2P                32
      IF(DOTP .LE. 0.D0) SGN=-1.D0                                      33
      N1=SGN*(X2-X0)/R                                                  34
      N2=SGN*(Y2-Y0)/R                                                  35
      CURV=SGN/R                                                        36
C
C         THE ANGLE OF THE CIRCULAR SEGMENT IS FOUND AND IF IT IS GREATER
C         THAT PI AN ERROR IS RETURNED.
C
      R1=DSQRT((X1-X0)*(X1-X0)+(Y1-Y0)*(Y1-Y0))                         37
      R2=DSQRT((X2-X0)*(X2-X0)+(Y2-Y0)*(Y2-Y0))                         38
      R3=DSQRT((X3-X0)*(X3-X0)+(Y3-Y0)*(Y3-Y0))                         39
C
C         (VI1,VI2) IS THE VECTOR FROM THE I-TH POINT TO THE ORGIN.
C
      V11=(X1-X0)/R1                                                    40
      V12=(Y1-Y0)/R1                                                    41
      V21=(X2-X0)/R2                                                    42
      V22=(Y2-Y0)/R2                                                    43
      V31=(X3-X0)/R3                                                    44
      V32=(Y3-Y0)/R3                                                    45
C
C         THET, THE1, THE2 ARE THE ANGLES FORMED BY THE SEGMENTS PT1-PT3,
C         PT1-PT2, AND PT2-PT3 RESPECTIVELY. NOTE THAT DARCOS RETURNS
C         VALUES FROM 0 TO PI.
C
      THET=DARCOS(V11*V31+V12*V32)                                      46
      THE1=DARCOS(V11*V21+V12*V22)                                      47
      THE2=DARCOS(V21*V31+V22*V32)                                      48
      T=DABS(THET-THE1-THE2)                                            49
      IF(T .GE. 1.D-4) GO TO 30                                         50
C
C         HALF THE ARC LENGTH BETWEEN PT1 AND PT2 IS ARC1 AND HALF THE ARC
C         LENGTH BETWEE  PT2 AND PT3 IS ARC2.
C
      ARC1=THE1*R*0.5D0                                                 51
      ARC2=THE2*R*0.5D0                                                 52
      RETURN                                                            53
   30 WRITE(6,50) X2,Y2                                                 54
      STOP 2                                                            55
   50 FORMAT(///5X,'***ERROR***'///10X,'THE POINTS AROUND ',F6.4,       56
     1 F6.4,' GIVE AN ARC OF MORE THAN PI*R')
      END                                                               57
```

DATE:08-29-79,13:33 OWNER:K3AK FILE:WAVE.S


```
                                                                              ISN
      SUBROUTINE WAVE(IO,ISNUM)                                                 1
C
C...  THIS ROUTINE COMPUTES THE WAVE AMPLITUDE (INCIDENT AND DIFFRACTED)
C...  ALONG THE SHIP LENGTH.
C
      COMMON RHO,GRAV,XLBP,HXLBP,XAXIS(21),YAXIS(21,25),                        2
     1ZAXIS(21,25),BEAM(21),
     2XI(50),XDEL(50),FROUD(8),U,XLAMDA(16),WAVEN(16),OMEGA(16),
     3AA(16),OMEGAE,TAU,TAUS,NSTA,NDIV,NDIV1
      COMMON /IWOD/ PHRE(21,50),PFRE(21,50),BO(21),PKVRE(21),DFVRE(21)          3
      COMMON /SIG/ SIGMA(50),SIGNE(50),SIGNDO(21),ALPHA,BX(50),                 4
     1SIGMAM(50),SIGNDM(50)
      COMMON /SHIPD/ X1(25),Y1(25),DEMAX(21),AN1(21,50),                        5
     1AN2(21,50),ACURV(21,50),ARC1(21,25),ARC2(21,25),IC(21,25),
     2NT(21),NT2(21),NT5(21)
      REAL*8 X1,Y1,DEMAX,AN1,AN2,ACURV,ARC1,ARC2                                6
      COMPLEX IM,SIGMA,SIGNE,SIGNDO,CONST,PND,ALPHA,SUM,ZETAND                  7
      DIMENSION ISNUM(21),ZETAND(21),ZMAG(21),ZANG(21)                          8
      IM=(0.0,1.0)                                                              9
      DO 999 ISTA=1,NSTA                                                       10
      NNT=NT(ISTA)                                                             11
      NNT2=NT2(ISTA)                                                           12
C...  COMPUTE CONSTANT FOR FIRST AND LAST STATIONS.
      IF(ISTA.EQ.1) CONST=-IM                                                  13
      IF(ISTA.NE.NSTA.OR.NNT2.GT.1) GO TO 3                                    14
      CONST=1.-ALPHA*(SUM(NNT2-1)-SIGMA(NDIV)*(G(XLBP,XLBP,XLBP)-              15
     1G(XLBP,XLBP,XI(NDIV)))/XDEL(NDIV))/AA(IO)
      CONST=-IM*CEXP(-IM*WAVEN(IO)*XLBP)*CONST                                 16
    3 IF(ISTA .NE. 1 .AND. ISTA .NE. NSTA)GO TO 8                              17
      IF(ISTA .EQ. NSTA .AND. NNT2 .GT. 1)GO TO 8                              18
      PFRE(ISTA,1)=1.                                                          19
      PHRE(ISTA,1)=0.0                                                         20
      GO TO 9                                                                  21
    8 CONST=IM*SIGND(ISNUM(ISTA))*CEXP(-IM*WAVEN(IO)*XAXIS(ISTA))/             22
     1(3.14159*BO(ISTA))
C...  COMPUTE NONDIMENSIONAL WAVE AMPLITUDE, MAGNITUDE AND PHASE.
    9 ZETAND(ISTA)=CONST*(PFRE(ISTA,1)+PHRE(ISTA,1))                           23
      ZMAG(ISTA)=CABS(ZETAND(ISTA))                                            24
      ZANG(ISTA)=ATAN2(AIMAG(ZETAND(ISTA)),REAL(ZETAND(ISTA)))                 25
     1*57.29578
  999 CONTINUE                                                                 26
      WRITE(6,2000)                                                            27
      WRITE(6,2010)  (XAXIS(ISTA),ZMAG(ISTA),ZANG(ISTA),ISTA=1,NSTA)           28
 2000 FORMAT(1H0,'  NONDIMENSIONAL WAVE AMPLITUDE',                            29
     1' (INCIDENT+DIFFRACTED) ALONG SHIP',/,
     2'   XAXIS   MAGNITUDE     PHASE')
 2010 FORMAT(3F10.4)                                                           30
      RETURN                                                                   31
      END                                                                      32
```

```
+------------------------------------------------------------+
| DATE:08-29-79,13:33  OWNER:K3AM  FILE:SOURCE.S              |
+------------------------------------------------------------+
```



```
                                                                         ISN
C
C     SUBROUTINE TO COMPUTE THE 3-D SOURCE STRENGTH DUE TO
C     INCIDENT WAVES USING LINEAR FIT
C
      SUBROUTINE SOURCE(IC,IFR)                                            1
      COMMON RHO,GRAV,XLBP,HXLBP,XAXIS(21),YAXIS(21,25),                   2
     1ZAXIS(21,25),BEAM(21),
     2XI(50),XDEL(50),FROUD(8),U,XLAMDA(16),WAVEN(16),OMEGA(16),
     3AA(16),OMEGAE,TAU,TAUS,NSTA,NDIV,NDIV1
      COMMON /SIG/ SIGMA(50),SIGND(50),SIGNDO(21),ALPHA,BX(50),            3
     1SIGMAM(50),SIGNDM(50)
      COMPLEX SIGMA,SIGND,SIGNDO,ALPHA                                     4
      COMPLEX IM,QT,BUG,DUM1,DUM2,SUM                                      5
      IM=(0.0,1.0)                                                         6
      SIGMA(1)=(0.0,0.0)                                                   7
C...  SET VALUE OF QT DEPENDING ON VALUE OF TAU.
      QT=IM*3.14159                                                        8
      IF(TAU.GT..25) QT=-2.221441                                          9
C...  SET VALUE OF ALPHA.
      ALPHA=SQRT(WAVEN(IC)*.159155/(1.+2.*TAUS))*(.707107,-.707107)       10
C...  MARCH ALONG SHIP FINDING SIGMA(J+1) USING VOLTERRA INTEGRAL EQN.
      DO 99 J=1,NDIV                                                      11
      J1=J+1                                                              12
      BUG=6.28318*BX(J1)*ALPHA                                            13
      DUM1=BUG*SIGMA(J)*(G(XI(J1),XI(J1),XI(J1))-G(XI(J1),XI(J1),XI(J)))  14
     1/XDEL(J)
      DUM2=QT*BX(J1)+2.-BUG*(G(XI(J),XI(J1),XI(J1))-                      15
     1G(XI(J),XI(J1),XI(J)))/XDEL(J)
      SIGMA(J1)=(-6.28318*BX(J1)*AA(IO)+BUG*SUM(J-1)-DUM1)/DUM2           16
   99 CONTINUE                                                            17
C...  SET NONDIMENSIONAL SIGMA.
      DO 20 I=1,NDIV1                                                     18
      SIGMAM(I)=CABS(SIGMA(I))                                            19
      SIGNDM(I)=SIGMAM(I)/AA(IO)                                          20
   20 SIGND(I)=SIGMA(I)/AA(IO)                                            21
      RETURN                                                              22
      END                                                                 23
```

```
                                                                          ISN
      COMPLEX FUNCTION SUM(JM1)
C                                                                           1
C...  THIS ROUTINE COMPUTES THE INTEGRAL FROM 0 TO XI(I-1) OF
C...  SIGMA/SQRT(X-XI)
C
CLOSE
      COMMON RHO,GRAV,XLBP,HXLBP,XAXIS(21),YAXIS(21,25),
     1ZAXIS(21,25),BEAM(21),                                                2
     2XI(50),XDEL(50),PRCUD(8),U,XLAMDA(16),WAVEN(16),OMEGA(16),
     3AA(16),OMEGAE,TAU,TAUS,NSTA,NDIV,NDIV1
      COMMON /SIG/ SIGMA(50),SIGND(50),SIGNDO(21),ALPHA,BX(50),
     1SIGMAM(50),SIGNDM(50)                                                 3
      COMPLEX SIGMA,SIGND,SIGNDO,ALPHA
      COMPLEX S                                                             4
      S=(0.0,0.0)                                                           5
      J1=JM1+2                                                              6
      IF(JM1 .EQ. 0)GO TO 11                                                7
      DO 10 I=1,JM1                                                         8
      I1=I+1                                                                9
      S=(SIGMA(I1)*(G(XI(I),XI(J1),XI(I1))-G(XI(I),XI(J1),XI(I)))-         10
     1SIGMA(I)*(G(XI(I1),XI(J1),XI(I1))-G(XI(I1),XI(J1),XI(I)))))/         11
     2XDEL(I)+S
   10 CONTINUE
   11 CONTINUE                                                             12
      SUM=S                                                                13
      RETURN                                                               14
      END                                                                  15
                                                                           16
```

```
                                                              ISN
      FUNCTION G(A,B,C)                                         1
C
C...  THIS ROUTINE IS USED IN COMPUTATION OF THE INTEGRAL
C...  (SIGMA(XI)/SQRT(X-XI)).
C
      G=(2*A-1.333333*B-.666667*C)*SQRT(B-C)                    2
      RETURN                                                    3
      END                                                       4
```

```
      SUBROUTINE TWODIN(COEF,YAXIS,ZAXIS,ISTA)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 V1,V3,YAXIS,ZAXIS
      REAL*8 N1,N2,K
      COMMON /SHIPD/ X1(25),Y1(25),DEMAX(21),AN1(21,50),
     1AN2(21,50),ACURV(21,50),ARC1(21,25),ARC2(21,25),IC(21,25),
     2NT(21),NT2(21),NT5(21)
      COMMON /TEST/ X,XP,Y,YP,K,BK(21),ICOUNT
      COMMON /DATA/ CURV,N1,N2
      DIMENSION YAXIS(21,25),ZAXIS(21,25)
      DIMENSION W(50),A(50,50),B(50),IPERM(50)
      DIMENSION PRE(50,50),FIM(50,50),BKC(50),X5(50),Y5(50),
     1W7(50),ACURV1(50),DEPRE(50),S1(50)
      DATA PI/3.14159265360/
C
C       FUNCTION TO COMPUTE THE RADIUS FROM (XP,YP) TO (X,Y)
C
      RAD(X,XP,Y,YP)=DSQRT((X-XP)**2+(Y-YP)**2)
C
C        SINCE THE WAVE NUMBER IS NON-DIMENSIONALIZED TO 1.0, THE
C        LENGTH SCALES ARE MODIFIED BY COEF. THE SYMMETRY OF THE HULL
C        IS USED TO GENERATE POINTS FOR THE ENTIER BODY.
C
      NNT=NT(ISTA)
      NNT2=NT2(ISTA)
      NNT5=NT5(ISTA)
      NNT1=NNT-1
      DO 15 I=1,NNT2
      ACURV1(I)=ACURV(ISTA,I)/COEF
      X5(I)=COEF*DBLE(YAXIS(ISTA,I))
      Y5(I)=COEF*DBLE(ZAXIS(ISTA,I))
C
C        W(I) IS THE ARCLENGTH SURROUNDING THE POINNT.
C
      W(I)=(ARC1(ISTA,I)+ARC2(ISTA,I))*COEF
      IF(I .EQ. NNT2) GO TO 15
      X5(2*NNT2-I)=-X5(I)
      Y5(2*NNT2-I)=Y5(I)
      W(2*NNT2-I)=W(I)
      ACURV1(2*NNT2-I)=ACURV1(I)
   15 CONTINUE
C
C       W7(I) IS TH ARCLENGTH BETWEEN THE I TH AND THE I+1 POINTS.
C
      DO 17 I=2,NNT5
      W7(I)=0.5D0*COEF*(ARC2(ISTA,I)+ARC1(ISTA,I+1))
   17 W7(NNT-I)=W7(I)
      W7(1)=0.5D0*COEF*(ARC1(ISTA,1)+ARC2(ISTA,1)+ARC1(ISTA,2))
      W7(NNT-1)=W7(1)
      K=1.
C
C       THE BODY BOUNDARY CONDITIONS ARE DETERMINED BY DCOS? AND
C       PLACED IN THE ARRAY B. (SUBROUTINE ACON IS ELIMINATED)
C
      DO 18 I=1,NNT
      B(I)=-N1*N2(-Y5(I))*AN2(ISTA,1)
   18 CONTINUE
C
```

```
C           THE GREEN'S FUNCTION, G(P,Q), AND ITS NORMAL DERIVATIVE
C           DG/DN ARE FOUND. NOTE THAT IF P=Q, G HAS ITS
C           SINGULAR NATURE SUBTRACTED.
C           FOR P .NE. Q, THE KO(R) TERM IS ALSO LEFT OFF
C
        DO 20 I=1,NNT2
        X=X5(I)
        Y=Y5(I)
        N1=AN1(ISTA,I)
        N2=AN2(ISTA,I)
        CURV=ACURV1(I)
        NNT7=NNT
        IF(I .EQ. NNT2) NNT7=NNT2
         DO 20 J=1,NNT7
        XP=X5(J)
        YP=Y5(J)
        CALL GSFUN(POTRE,POTIM,PNRE,PNIM)
        PRE(I,J)=POTRE
        PNRE=W(J)*PNRE
C
C           THE INFLUENCE OF THE SOURCE AT P ON THE POINT P IS PI.
C
        IF(I.EQ.J) PNRE=PNRE-PI
     19 A(I,J)=PNRE
        IF(I .EQ. NNT2 .AND. J .EQ. NNT2) GO TO 20
        A(NNT-I+1,NNT-J+1)=A(I,J)
        PRE(NNT-I+1,NNT-J+1)=PRE(I,J)
     20 CONTINUE
C
C           A IS A NNT*NNT COMPLEX MATRIX THAT CONTAINS THE COEFFICIENTS
C           TO THE SET OF SIMULTANEOUS LINEAR EQUATIONS OF WHICH THE
C           SOURCE STRENGTH IS THE SOLUTION.
C
        CALL DLUD(NNT,50,A,50,A,IPERM)
        IF(IPERM(NNT).NE.0) GO TO 25
        WRITE(6,110)
        STOP
     25 CALL DBS(NNT,50,A,IPERM,B)
C
C           THE SOUCE DISTRIBUTION IS NOW FOUND IN B.
C
        DO 70 I=1,NNT
        X=X5(I)
        Y=Y5(I)
        IFST=0
C
C           INTEGRATION OF THE PRODUCT B*G(P,Q) TO FIND
C           THE POTENTIAL AT P.
C
        DO 40 J=1,NNT
        IF(I .EQ. J) GO TO 37
        R1(J)=RAD(X,X5(J),Y,Y5(J))
        CALL BESK(R1(J),0,BKO(J),IER)
        GO TO 40
     37 BKO(J)=0.D0
        B1(J)=0.D0
     40 CONTINUE
        SUM1=0.D0
```

```
C
C        TRAPEZOIDAL INTEGRATION FORMULA USED TO INTEGRATE
C        THE NON-SINGULAR PART OF G(P,Q).
C
      DO 60 J=1,NNT1
      RP1=RAD(X,X5(J),Y,-Y5(J))
      RP2=RAD(X,X5(J+1),Y,-Y5(J+1))
C
C        A CHECK IS MADE TO SEE IF AT EITHER END POINT P=Q
C
      IF(RP1.LE. 1.D-5) GO TO 50
      IF(RP2.LE. 1.D-5) GO TO 57
      IF(J+1 .EQ. I .OR. J .EQ. I) GO TO 45
      SUM1=SUM1+W7(J)*(B(J)*(BKO(J)+PRE(I,J))+
     1             B(J+1)*(BKO(J+1)+PRE(I,J+1)))
      GO TO 60
   45 SUM1=SUM1+W7(J)*(B(J)*PRE(I,J)+B(J+1)*PRE(I,J+1))
      GO TO 60
   50 CALL BESK(RP2,0,BKOP,IER)
      SUM1=SUM1+W7(J)*(B(J)*PRE(I,J)+B(J+1)*(PRE(I,J+1)-BKOP))
      ITEST=1
      GO TO 60
   57 CALL BESK(RP1,0,BKOP,IER)
      SUM1=SUM1+W7(J)*(B(J)*(PRE(I,J)-BKOP)+B(J+1)*PRE(I,J+1))
      ITEST=1
      GO TO 60
   60 CONTINUE
C
C        THE SINGULAR PART OF G(P,Q) IS ADDED, ASSUMING A
C        LINEAR SOURCE DISTRIBUTION.
C
      IF(I .EQ. 1) GO TO 64
      IF(I .EQ. NNT) GO TO 67
      ARG1=R1(I-1)
      ARG2=R1(I+1)
      CALL BIK(SNGL(ARG1),V1)
      CALL BESK(ARG1,1,VI2,IER)
      CALL BIK(SNGL(ARG2),V3)
      CALL BESK(ARG2,1,VI4,IER)
      A1R=B(I)
      B1R=(B(I-1)-A1R)/R1(I-1)
      B2R=(B(I+1)-A1R)/R1(I+1)
      VI1=DBLE(V1)
      VI3=DBLE(V3)
      DPHBE(I)=SUM1+A1R*VI1+B1R*(1.DO-ARG1*VI2)
     1         +A1R*VI3+B2R*(1.DO-ARG2*VI4)
       GO TO 70
   64 ARG2=R1(2)
      A1R=B(1)
      B2R=(B(2)-A1R)/R1(2)
      GO TO 69
   67 ARG2=R1(NNT-1)
      A1P=B(NNT)
      B2P=(B(NNT-1)-A1R)/R1(NNT-1)
   69 CALL BIK(SNGL(ARG2),V3)
      CALL BESK(ARG2,1,VI4,IER)
      VI3=DBLE(V3)
      DPHBE(I)=SUM1+(1.DO+DFLOAT(ITEST))*(A1R*VI3+B2R*(1.DO
```

```
                                                  ISN
     1                          -ARG2*VIQ))
70  CONTINUE
    CALL TWDATA(B,W7,X5,Y5,CPHBE,COEF,AN1,AN2,BPMAX,NNT,      120
   1NNT1,NNT2,ISTA)                                            121
110 FORMAT(' ******** ERROR *********',//5X,'THE SOURCE MATRIX '
   1 ,'SINGULAR')                                              122
    RETURN
    END                                                        123
                                                               124
```

```
      SUBROUTINE TWDATA(B,W7,X5,Y5,DPHRE,COEF,AN1,AN2,DEMAX,NNT,
     1NNT1,NNT2,ISTA)
C
C     +---------------------------------------------+
C     |                                             |
C     |    THIS ROUTINE COMPUTES THE 2-D POTENTIAL  |
C     |  EXCITING FORCES AND BO.  IT ALSO CONVERTS  |
C     |  ISOESH DOUBLE PRECISION TO SINGLE PRECISION.|
C     |                                             |
C     +---------------------------------------------+
C
      COMMON /ISOD/ PHRE(21,50),PFRE(21,50),BO(21),FKVRE(21),DPVRE(21)
      REAL*8 B(50),W7(50),X5(50),Y5(50),DPHRE(50),COEF,AN1(21,50)
     1,AN2(21,50),DEMAX(21),AO,SUMF,SUMD,FACT,DPFRE(50)
      AO=0.D0
      DO 10 I=1,NNT
      PHRE(ISTA,I)=DPHRE(I)
      DPFRE(I)=DEXP(-Y5(I))
   10 PFRE(ISTA,I)=DPFRE(I)
      DO 15 I=1,NNT1
C...  COMPUTE AO BY TRAPEZOIDAL INTEGRATION AROUND SECTION.
   15 AO=AO+W7(I)*(B(I)*DPFRE(I)+B(I+1)*DPFRE(I+1))
C BO IS TWICE AO AND CHANGE TO SINGLE PRECISION
      BO(ISTA)=2.D0*AO
      SUMF=0.0D0
      SUMD=0.0D0
      DO 20 I=1,NNT1
C...  COMPUTE 2-D EXCITING FORCES
C...  BY TRAPEZOIDAL INTEGRATION OF PRESSURE AROUND SECTION.
      SUMF=SUMF+W7(I)*(AN2(ISTA,I)*DPFRE(I)+AN2(ISTA,I+1)*DPFRE(I+1))
      SUMD=SUMD+W7(I)*(AN2(ISTA,I)*DPHRE(I)+AN2(ISTA,I+1)*DPHRE(I+1))
   20 CONTINUE
      FACT=1.D0/(DEMAX(ISTA)*COEF)
      FKVRE(ISTA)=SUMF*FACT
      DPVRE(ISTA)=SUMD*FACT
      RETURN
      END
```

```
                                                                        ISN
      SUBROUTINE INTRPL(IU,L,X,Y,N,U,V)                                   1
C
C...  Interpolation of a Single Valued Function
C
C        This routine interpolates, from values of the function
C     given as ordinates of input data points in an X-Y plane
C     and for a given set of X values (abscissas), the values of
C     a single-valued function Y=Y(X).
C
C...  The input parameters are
C
C        IU = Logical unit number of standard output unit
C        L  = Number of input data points
C             (Must be two or greater)
C        X  = Array of dimension L storing the X values
C             (Abscissas) of input data points.
C             (In ascending order)
C        Y  = Array of dimension L storing the Y values
C             (Ordinates) of input data points.
C        N  = Number of points at which interpolation of the
C             Y value (Ordinate) is desired.
C             (Must be one or greater)
C        U  = Array of dimension N storing the X values
C             (Abscissas) of desired points.
C
C...  The output parameter is
C
C        V  = Array of dimension N where the interpolated Y
C             values (Ordinates) are to be displayed.
C
C...  Declaration Statements
C
      DIMENSION X(L),Y(L),U(N),V(N)                                       2
      EQUIVALENCE (P0,X3),(Q0,Y3),(Q1,T3)                                 3
      REAL M1,M2,M3,M4,M5                                                 4
      EQUIVALENCE (UK,DX),(IMN,X2,A1,M1),(IMX,X5,A5,M5),                  5
     *            (J,SW,SA,(Y2,W2,W4,Q2),(Y5,W3,Q3)
C...  Preliminary Processing
   10 LO=L                                                                6
      LM1=LO-1                                                            7
      LM2=LM1-1                                                           8
      LP1=LO+1                                                            9
      NO=N                                                               10
      IF(LM2 .LT. 0)GO TO 90                                             11
      IF(NO .LE. 0)GO TO 91                                              12
      DO 11 I=2,LO                                                       13
        IF(X(I-1)-X(I))11,95,96                                          14
   11 CONTINUE                                                           15
      IPV=0                                                              16
C...  Main DO - loop
      DO 80 K=1,NO                                                       17
        UK=U(K)                                                          18
C...  Routine to locate the desired point
   20   IF(LM2 .EQ. 0)GO TO 27                                           19
        IF(UK .GE. X(LO))GO TO 26                                        20
        IF(UK .LT. X(1))GO TO 25                                         21
        IMN=2                                                            22
        IMX=LO                                                           23
```

```
                                                      ISN
  21     I=(INN+INX)/2                                 24
         IF(UK .GE. X(I))GO TO 23                      25
  22     INX=I                                         26
         GO TO 24                                      27
  23     INN=I+1                                       28
  24     IF(INX .GT. INN)GO TO 21                      29
         I=INX                                         30
         GO TO 30                                      31
  25     I=1                                           32
         GO TO 30                                      33
  26     I=LP1                                         34
         GO TO 30                                      35
  27     I=2                                           36
C...  Check if I = IPV
  30     IF(I .EQ. IPV)GO TO 70                        37
         IPV=I                                         38
C...  Routines to pick up necessary X and Y values and
C...              to estimate them if necessary.
  40     J=I                                           39
         IF(J .EQ. 1)J=2 .                             40
         IF(J .EQ. LP1)J=L0                            41
         X3=X(J-1)                                     42
         Y3=Y(J-1)                                     43
         X4=X(J)                                       44
         Y4=Y(J)                                       45
         A3=X4-X3                                      46
         M3=(Y4-Y3)/A3                                 47
         IF(LM2 .EQ. 0)GO TO 43                        48
         IF(J .EQ. 2)GO TO 41                          49
         X2=X(J-2)                                     50
         Y2=Y(J-2)                                     51
         A2=X3-X2                                      52
         M2=(Y3-Y2)/A2                                 53
         IF(J .EQ. L0)GO TO 42                         54
  41     X5=X(J+1)                                     55
         Y5=Y(J+1)                                     56
         A4=X5-X4                                      57
         M4=(Y5-Y4)/A4                                 58
         IF(J .EQ. 2)M2=M3+M3-M4                       59
         GO TO 45                                      60
  42     M4=M3+M3-M2                                   61
         GO TO 45                                      62
  43     M2=M3                                         63
         M4=M3                                         64
  45     IF(J .LE. 3)GO TO 46                          65
         A1=X2-X(J-3)                                  66
         M1=(Y2-Y(J-3))/A1                             67
         GO TO 47                                      68
  46     M1=M2+M2-M3                                   69
  47     IF(J .GE. LM1)GO TO 48                        70
         A5=X(J+2)-X5                                  71
         M5=(Y(J+2)-Y5)/A5                             72
         GO TO 50                                      73
  48     M5=M4+M4-M3                                   74
C...  Numerical Differentiation
  50     IF(I .EQ. LP1)GO TO 52                        75
         W2=ABS(M4-M3)                                 76
         W3=ABS(M2-M1)                                 77
```

```
      SW=W2+W3
      IF(SW .NE. 0.0)GO TO 51
      W2=0.5
      W3=0.5
      SW=1.0
   51 T3=(W2*M2+W3*M3)/SW
      IF(I .EQ. 1)GO TO 54
   52 W3=ABS(M5-M4)
      W4=ABS(M3-M2)
      SW=W3+W4
      IF(SW .NE. 0.0)GO TO 53
      W3=0.5
      W4=0.5
      SW=1.0
   53 T4=(W3*M3+W4*M4)/SW
      IF(I .NE. LP1)GO TO 60
      T3=T4
      SA=A2+A3
      T4=0.5*(M4+M5-A2*(A2-A3)*(M2-M3)/(SA*SA))
      X3=X4
      Y3=Y4
      A3=A2
      M3=M4
      GO TO 60
   54 T4=T3
      SA=A3+A4
      T3=0.5*(M1+M2-A4*(A3-A4)*(M3-M4)/(SA*SA))
      X3=X3-A4
      Y3=Y3-M2*A4
      A3=A4
      M3=M2
C...  Determination of the coefficients
   60 Q2=(2.0*(M3-T3)+M3-T4)/A3
      Q3=(-M3-M3+T3+T4)/(A3*A3)
C...  Computation of the polynomial
   70 DX=UK-P0
   80 V(K)=Q0+DX*(Q1+DX*(Q2+DX*Q3))
      RETURN
C...  Error exit
   90 WRITE(IU,2090)
      GO TO 99
   91 WRITE(IU,2091)
      GO TO 99
   95 WRITE(IU,2095)
      GO TO 97
   96 WRITE(IU,2096)
   97 WRITE(IU,2097)I,X(I)
   99 WRITE(IU,2099)LO,NO
      RETURN
C...  Format Statements
 2090 FORMAT(1X/'   ***   I=1 OR LESS.'/)
 2091 FORMAT(1X/'   ***   N=0 OR LESS.'/)
 2095 FORMAT(1X/'   ***   Identical X values.'/)
 2096 FORMAT(1X/'   ***   X values out of sequence'/)
 2097 FORMAT(1X,'   I= ',I7,'   X(I)= ',E12.3)
 2099 FORMAT(1X,'   L= ',I7,'   N   = ',I7/
     *      5X,'Error detected in routine INTERPL')
      END
```

```
      C
      C************************************
      C
      C DECK  DOL4
      C                                                               DOL4  10
      C   ............................................................DOL4  20
      C                                                               DOL4  30
      C      SUBROUTINE DOL4                                          DOL4  40
      C                                                               DOL4  50
      C      PURPOSE                                                  DOL4  60
      C         TO COMPUTE INTEGRAL(EXP(-X)*FCT(X), SUMMED OVER X     DOL4  70
      C                         FROM 0 TO INFINITY).                  DOL4  80
      C                                                               DOL4  90
      C      USAGE                                                    DOL4 100
      C         CALL DOL4 (FCT,Y)                                     DOL4 110
      C         PARAMETER FCT REQUIRES AN EXTERNAL STATEMENT          DOL4 120
      C                                                               DOL4 130
      C      DESCRIPTION OF PARAMETERS                                DOL4 140
      C         FCT    - THE NAME OF AN EXTERNAL DOUBLE PRECISION FUNCTION DOL4 150
      C                  SUBPROGRAM USED.                             DOL4 160
      C         Y      - THE RESULTING DOUBLE PRECISION INTEGRAL VALUE. DOL4 170
      C                                                               DOL4 180
      C      REMARKS                                                  DOL4 190
      C         NONE                                                  DOL4 200
      C                                                               DOL4 210
      C      SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED            DOL4 220
      C         THE EXTERNAL DOUBLE PRECISION FUNCTION SUBPROGRAM FCT(X) DOL4 230
      C         MUST BE FURNISHED BY THE USER.                        DOL4 240
      C                                                               DOL4 250
      C      METHOD                                                   DOL4 260
      C         EVALUATION IS DONE BY MEANS OF 4-POINT GAUSSIAN-LAGUERRE DOL4 270
      C         QUADRATURE FORMULA, WHICH INTEGRATES EXACTLY,         DOL4 280
      C         WHENEVER FCT(X) IS A POLYNOMIAL UP TO DEGREE 7.       DOL4 290
      C         FOR REFERENCE, SEE                                    DOL4 300
      C         SHAO/CHEN/FRANK, TABLES OF ZEROS AND GAUSSIAN WEIGHTS OF DOL4 310
      C         CERTAIN ASSOCIATED LAGUERRE POLYNOMIALS AND THE RELATED DOL4 320
      C         GENERALIZED HERMITE POLYNOMIALS, IBM TECHNICAL REPORT DOL4 330
      C         TR00.1100 (MARCH 1964), PP.24-25.                     DOL4 340
      C                                                               DOL4 350
      C   ............................................................DOL4 360
      C                                                               DOL4 370
          SUBROUTINE DOL4(FCT,Y)                                      DOL4 380     1
      C                                                               DOL4 390
      C                                                               DOL4 400
          DOUBLE PRECISION X,Y,FCT                                    DOL4 410     2
      C                                                               DOL4 420
          X=.93950709123011330D1                                      DOL4 430     3
          Y=.53929470556132745D-3*FCT(X)                              DOL4 440     4
          X=.45366202969211280D1                                      DOL4 450     5
          Y=Y+.38887908515005384D-1*FCT(X)                            DOL4 460     6
          X=.17457611011583466D1                                      DOL4 470     7
          Y=Y+.35741869243779969D0*FCT(X)                             DOL4 480     8
          X=.32254768961939231D0                                      DOL4 490     9
          Y=Y+.60315410434163360D0*FCT(X)                             DOL4 500    10
          RETURN                                                      DOL4 510    11
          END                                                         DOL4 520    12
```

```
C DECK  DOL8
C                                                                    DOL8  10
C     ...............................................................DOL8  20
C                                                                    DOL8  30
C        SUBROUTINE DOL8                                             DOL8  40
C                                                                    DOL8  50
C        PURPOSE                                                     DOL8  60
C           TO COMPUTE INTEGRAL(EXP(-X)*FCT(X), SUMMED OVER X         DOL8  70
C                              FROM 0 TO INFINITY).                  DOL8  80
C                                                                    DOL8  90
C        USAGE                                                       DOL8 100
C           CALL DOL8 (FCT,Y)                                        DOL8 110
C           PARAMETER FCT REQUIRES AN EXTERNAL STATEMENT             DOL8 120
C                                                                    DOL8 130
C        DESCRIPTION OF PARAMETERS                                   DOL8 140
C           FCT    - THE NAME OF AN EXTERNAL DOUBLE PRECISION FUNCTION DOL8 150
C                    SUBPROGRAM USED.                                DOL8 160
C           Y      - THE RESULTING DOUBLE PRECISION INTEGRAL VALUE.  DOL8 170
C                                                                    DOL8 180
C        REMARKS                                                     DOL8 190
C           NONE       .                                             DOL8 200
C                                                                    DOL8 210
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED               DOL8 220
C           THE EXTERNAL DOUBLE PRECISION FUNCTION SUBPROGRAM FCT(X) DOL8 230
C           MUST BE FURNISHED BY THE USER.                           DOL8 240
C                                                                    DOL8 250
C        METHOD                                                      DOL8 260
C           EVALUATION IS DONE BY MEANS OF 8-POINT GAUSSIAN-LAGUERRE DOL8 270
C           QUADRATURE FORMULA, WHICH INTEGRATES EXACTLY,            DOL8 280
C           WHENEVER FCT(X) IS A POLYNOMIAL UP TO DEGREE 15.         DOL8 290
C           FOR REFERENCE, SEE                                       DOL8 300
C           SHAO/CHEN/FRANK, TABLES OF ZEROS AND GAUSSIAN WEIGHTS OF DOL8 310
C           CERTAIN ASSOCIATED LAGUERRE POLYNOMIALS AND THE RELATED  DOL8 320
C           GENERALIZED HERMITE POLYNOMIALS, IBM TECHNICAL REPORT    DOL8 330
C           TR00.1100 (MARCH 1964), PP.24-25.                        DOL8 340
C                                                                    DOL8 350
C     ...............................................................DOL8 360
C                                                                    DOL8 370
      SUBROUTINE DOL8(FCT,Y)                                         DOL8 380     1
C                                                                    DOL8 390
C                                                                    DOL8 400
      DOUBLE PRECISION X,Y,FCT                                       DOL8 410     2
C                                                                    DOL8 420
      X=.22863131736889264D2                                         DOL8 430     3
      Y=.10480011748715104D-8*FCT(X)                                 DOL8 440     4
      X=.15740678641278005D2                                         DOL8 450     5
      Y=Y+.84857467162725315D-6*FCT(X)                               DOL8 460     6
      X=.10758516010180995D2                                         DOL8 470     7
      Y=Y+.90765087733582131D-4*FCT(X)                               DOL8 480     8
      X=.70459054023934657D1                                         DOL8 490     9
      Y=Y+.27945362352256725D-2*FCT(X)                               DOL8 500    10
      X=.42667001702876588D1                                         DOL8 510    11
      Y=Y+.33343492261215652D-1*FCT(X)                               DOL8 520    12
      X=.22510866298661307D1                                         DOL8 530    13
      Y=Y+.17579498663717181D0*FCT(X)                                DOL8 540    14
      X=.90370177679537990D0                                         DOL8 550    15
      Y=Y+.41878678081434296D0*FCT(X)                                DOL8 560    16
      X=.17027963230510100D0                                         DOL8 570    17
```

```
Y=Y+.369183583416375300*CT(Z)
RETURN
END
```

PAGE DATE ... 11:33 OWNER: ... FILE: HELMSUB.S 1	<PAGE 28>
	ISN
DOLB 549	18
DOLB	19
DOLB 600	20

```
C DECK  DL12
C                                                                       DL12  10
C     ..................................................................DL12  20
C                                                                       DL12  30
C        SUBROUTINE DQL12                                               DL12  40
C                                                                       DL12  50
C        PURPOSE                                                        DL12  60
C           TO COMPUTE INTEGRAL(EXP(-X)*FCT(X), SUMMED OVER X           DL12  70
C                               FROM 0 TO INFINITY).                    DL12  80
C                                                                       DL12  90
C        USAGE                                                          DL12 100
C           CALL DQL12  (FCT,Y)                                         DL12 110
C           PARAMETER FCT REQUIRES AN EXTERNAL STATEMENT                DL12 120
C                                                                       DL12 130
C        DESCRIPTION OF PARAMETERS                                      DL12 140
C           FCT    - THE NAME OF AN EXTERNAL DOUBLE PRECISION FUNCTION  DL12 150
C                    SUBPROGRAM USED.                                   DL12 160
C           Y      - THE RESULTING DOUBLE PRECISION INTEGRAL VALUE.     DL12 170
C                                                                       DL12 180
C        REMARKS                                                        DL12 190
C           NONE                                                        DL12 200
C                                                                       DL12 210
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                  DL12 220
C           THE EXTERNAL DOUBLE PRECISION FUNCTION SUBPROGRAM FCT(X)    DL12 230
C           MUST BE FURNISHED BY THE USER.                              DL12 240
C                                                                       DL12 250
C        METHOD                                                         DL12 260
C           EVALUATION IS DONE BY MEANS OF 12-POINT GAUSSIAN-LAGUERRE   DL12 270
C           QUADRATURE FORMULA, WHICH INTEGRATES EXACTLY,               DL12 280
C           WHENEVER FCT(X) IS A POLYNOMIAL UP TO DEGREE 23.            DL12 290
C           FOR REFERENCE, SEE                                          DL12 300
C           SHAO/CHEN/FRANK, TABLES OF ZEROS AND GAUSSIAN WEIGHTS OF    DL12 310
C           CERTAIN ASSOCIATED LAGUERRE POLYNOMIALS AND THE RELATED     DL12 320
C           GENERALIZED HERMITE POLYNOMIALS, IBM TECHNICAL REPORT       DL12 330
C           TR00.1100 (MARCH 1964), PP.24-25.                           DL12 340
C                                                                       DL12 350
C     ..................................................................DL12 360
C                                                                       DL12 370
      SUBROUTINE DQL12(FCT,Y)                                           DL12 380    1
C                                                                       DL12 390
C                                                                       DL12 400
      DOUBLE PRECISION X,Y,FCT                                          DL12 410    2
C                                                                       DL12 420
      X=.37099121044466920D2                                            DL12 430    3
      Y=.81480774674267420D-15*FCT(X)                                   DL12 440    4
      X=.28487967250984000D2                                            DL12 450    5
      Y=Y+.30616016350350208D-11*FCT(X)                                 DL12 460    6
      X=.22151090379397006D2                                            DL12 470    7
      Y=Y+.13423910305150041D-8*FCT(X)                                  DL12 480    8
      X=.17116855187462256D2                                            DL12 490    9
      Y=Y+.16684938765409103D-6*FCT(X)                                  DL12 500   10
      X=.13006054993306348D2                                            DL12 510   11
      Y=Y+.8365055856819799D-5*FCT(X)                                   DL12 520   12
      X=.9621316842456867D1                                             DL12 530   13
      Y=Y+.20323159266299939D-3*FCT(X)                                  DL12 540   14
      X=.68445254531151773D1                                            DL12 550   15
      Y=Y+.26639735418653159D-2*FCT(X)                                  DL12 560   16
      X=.45992276394183485D1                                            DL12 570   17
```

```
                                                          ISN
Y=Y+.20102381154634097D-1*FCT(X)        DL12 580          18
X=.29337513377435072D1                  DL12 590          19
Y=Y+.50444922221160893D-1*FCT(X)        DL12 600          20
X=.15126302657726418D1                  DL12 610          21
Y=Y+.24408201131987756DO*FCT(X)         DL12 620          22
X=.61175748451513067DO                  DL12 630          23
Y=Y+.37775927587313798DO*FCT(X)         DL12 640          24
X=.11572211735802068DO                  DL12 650          25
Y=Y+.26473137105544319DO*FCT(X)         DL12 660          26
RETURN                                  DL12 670          27
END                                     DL12 680          28
```

-48-

```
C NAASA 10.1.022 NATSEI   FTN   06-24-75     THE UNIV OF MICH COMP CTR  NATS   1
C                                                                       NATS   2
C      ------- FUNPACK ------- IBM S/360 ------ LONG PRECISION --------- NATS   3
C                                                                       NATS   4
       SUBROUTINE NATSEI(ARG,RESULT,INT)                                NATS   5     1
C                                                                       NATS   6
       REAL*8 A(6),B(6),C(8),D(8),E(8),F(8),P1(9),Q1(9),P2(9),Q2(8)     NATS   7     2
     1        ,P3(10),Q3(9),P4(10),Q4(9),P0(6),Q0(6),PX(9),QX(9)        NATS   8
     2        ,FCN(3),FRAC,SUMP,SUMQ,T,W,X,X0,XX0,X01,X02,XMX0,Y        NATS   9
     3        ,DEXP40,XINF,XMAX,XMIN,EI,ARG,RESULT,DABS,DEXP,DLOG       NATS  10
       INTEGER I,IFCN,INT,JK                                            NATS  11     3
C                                                                       NATS  12
C      THIS  FUNPACK  PACKET CONTAINS THREE FUNCTION TYPE SUBPROGRAMS   NATS  13
C      DEI, DPEONE, AND  DEXPEI, AND ONE SUBROUTINE TYPE SUBPROGRAM,    NATS  14
C      NATSEI.  THE PACKET COMPUTES LONG PRECISION VALUES OF THE        NATS  15
C      EXPONENTIAL INTEGRALS                                            NATS  16
C                                                                       NATS  17
C              EI(X),  E-SUB-1(X) = -EI(-X),  AND  EXP(-X)*EI(X)        NATS  18
C                                                                       NATS  19
C      WHERE                                                            NATS  20
C                                                                       NATS  21
C             INTEGRAL (FROM T=-INFINITY TO T=X) (EXP(T)/T),  X .GT. 0, NATS  22
C      EI(X) =                                                          NATS  23
C             -INTEGRAL (FROM T=-X TO T=INFINITY) (EXP(-T)/T),          NATS  24
C                                                      X .LT. 0,        NATS  25
C                                                                       NATS  26
C      AND WHERE THE FIRST INTEGRAL IS A PRINCIPAL VALUE INTEGRAL.      NATS  27
C      THE FORTRAN CALLING STATEMENTS FOR THE PRIMARY ENTRIES TO THIS   NATS  28
C      PACKET ARE                                                       NATS  29
C                                                                       NATS  30
C                  Y = DEI(X),           WHERE  X .NE. 0,               NATS  31
C                                                                       NATS  32
C                  Y = DPEONE(X),        WHERE  X .GT. 0,               NATS  33
C      AND                                                              NATS  34
C                  Y = DEXPEI(X),        WHERE  X .NE. 0,               NATS  35
C                                                                       NATS  36
C      AND WHERE THE ENTRY POINTS CORRESPOND TO THE FUNCTIONS EI(X),    NATS  37
C      E-SUB-1(X) AND EXP(-X)*EI(X), RESPECTIVELY.  THE ROUTINE  NATSEI NATS  38
C      IS INTENDED FOR INTERNAL PACKET USE ONLY, ALL COMPUTATIONS WITHIN NATS  39
C      THE PACKET BEING CONCENTRATED IN THIS ROUTINE.  THE FUNCTION     NATS  40
C      SUBPROGRAMS INVOKE  NATSEI  WITH THE FORTRAN STATEMENT           NATS  41
C           CALL NATSEI(ARG,RESULT,INT)                                 NATS  42
C      WHERE THE PARAMETER USAGE IS AS FOLLOWS                          NATS  43
C                                                                       NATS  44
C        FUNCTION                PARAMETERS FOR  NATSEI                 NATS  45
C          CALL               ARG            RESULT         INT         NATS  46
C        DEI(X)             X .NE. 0         EI(X)           1          NATS  47
C        DPEONE(X)          X .GT. 0        -EI(-X)          2          NATS  48
C        DEXPEI(X)          X .NE. 0         EXP(-X)*EI(X)   3          NATS  49
C                                                                       NATS  50
C      THE MAIN COMPUTATION INVOLVES EVALUATION OF RATIONAL CHEBYSHEV   NATS  51
C      APPROXIMATIONS PUBLISHED IN MATH. COMP. 22, 641-649(1968), AND   NATS  52
C      MATH. COMP. 23, 289-303(1969) BY CODY AND THACHER.               NATS  53
C                                                                       NATS  54
C      OTHER SUBPROGRAMS REQUIRED                                       NATS  55
C                                                                       NATS  56
C      FROM  FUNPACK  -  FCNFCN  AND, OPTIONALLY, MCNFRR  (BOTH         NATS  57
C                          CONTAINED IN THE  FUNPACK  PACKET  MCNEPR)   NATS  58
```

```
C     OTHER          -  DABS, DLOG, DEXP
C
C     ACCURACY
C
C     THIS SUBROUTINE IS DESIGNED TO GIVE NEAR OPTIMAL ACCURACY
C     IN LONG PRECISION ARITHMETIC ON AN IBM S/360 COMPUTER.
C     THE FOLLOWING TABLE GIVES THE RESULTS OF RANDOM ARGUMENT
C     ACCURACY TESTS FOR THE  DEI  ENTRY.
C
C      ARGUMENT        FREQUENCY OF BIT ERRORS    MAX.       RMS
C       RANGE           NO. OF BITS IN ERROR       REL.       REL.
C                        0    1    2    3  OTHER    ERROR      ERROR
C
C     (-50.0, -4.0)   254  890  397  350  109   0.460D-15  0.141D-15
C     (-4.0, -1.0)     53  414  896  241  396   0.631D-15  0.247D-15
C     (-1.0,  0.0)    275  754  758   50  163   0.444D-15  0.138D-15
C     ( 0.0,  6.0)    592  605  403  338   62   0.116D-14  0.120D-15
C     ( 6.0, 12.0)    419  668  603  277   34   0.466D-15  0.121D-15
C     (12.0, 24.0)    511  677  490  242   60   0.41[illegible]D-15  0.10[illegible]D-15
C     (24.0, 100.)    529  566  452  334   79   0.251D-15  0.101D-15
C       ------------
C
C     ERROR MONITORING
C
C
C        THE FOLLOWING TABLE INDICATES THE TYPES OF ERROR
C     THAT MAY BE ENCOUNTERED IN THIS ROUTINE AND THE FUNCTION
C     VALUE SUPPLIED IN EACH CASE (XINF IS THE LARGEST POSITIVE
C     MACHINE NUMBER).
C
C     ERROR   ERROR        ARGUMENT        FUNCTION VALUES FOR
C      NO.                  RANGE        DEI      DEXPEI     DPEONE
C       1    UNDERFLOW    X .LT. XMIN    0          -          0
C       2    OVERFLOW     X .GE. XMAX   XINF        -          -
C       3    ILLEGAL X      X = 0      -XINF      -XINF       XINF
C       4    ILLEGAL X     X .LT. 0      -          -       USE ABS(X)
C
C     ALL ERROR PROCESSING IS HANDLED THROUGH THE SEPARATE  FUNPACK
C     PACKET  MONERR.  THE DEFAULT ERROR PROCEDURE IS TO PRINT
C     AN ERROR MESSAGE ONLY FOR THE FIRST ERROR ENCOUNTERED.  THE
C     FREQUENCY OF EACH TYPE OF ERROR IS ACCUMULATED, AND EXECUTION
C     IS CONTINUED.  FOR THE  DEI  PACKET,  MONERR  MAY BE INVOKED
C     TO ALTER THESE CONVENTIONS, OR TO INTERROGATE THE ERROR STATUS,
C     BY THE FORTRAN STATEMENT
C                 JK = MONERR(KK,1),
C     WHERE  JK  AND  KK  ARE INTEGERS, RELATED ACCORDING TO THE
C     FOLLOWING TABLE.  EC-N DENOTES THE ERROR COUNT FOR ERROR
C     NUMBER  N  IN THE ABOVE TABLE, AND  --  DENOTES NO CHANGE
C     FROM THE PREVIOUS SETTING.
C
C                                           SUBSEQUENT ACTION
C          KK     JK        ERROR           UPON FINDING ERROR
C                           COUNTS           PRINT     TRACEBACK
C                                            MSSG.     AND STOP
C        .LT. -1   0         --               --         --
C           -1     KK        --               YES        YES
C            0     KK     RESET TO 0        1 ONLY       NO
C            1     KK        --             EVERY TIME   NO
C            2     KK        --               NO         NO
```

```
C           3      *      --          --          --          NATS 117
C           4     EC-1    --          --          --          NATS 118
C           5     EC-2    --          --          --          NATS 119
C           6     EC-3    --          --          --          NATS 120
C           7     EC-4    --          --          --          NATS 121
C         .GT. 7   0      --          --          --          NATS 122
C                                                             NATS 123
C      * THE RETURN FOR KK=3 IS THE ERROR NUMBER N (FROM THE  NATS 124
C        PREVIOUS TABLE) FOR THE LATEST ERROR ENCOUNTERED.   NATS 125
C        AT THE SAME TIME, THE INTERNAL ERROR INDICATOR IS   NATS 126
C        RESET TO 0.  A RETURN OF 0 INDICATES NO ERROR HAS   NATS 127
C        OCCURRED.                                            NATS 128
C                                                             NATS 129
C        ----------                                           NATS 130
C                                                             NATS 131
C     QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW  NATS 132
C     APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY  NATS 133
C                                                             NATS 134
C     ------------------------------------------------------------ NATS 135
      DATA DEXP40/Z4F34441A72F2E5D5/,XINF/Z7FFFFFFFFFFFFFFF/  NATS 136      4
     1      ,XMAX/Z42B3DC12A2B5787F/,XMIN/ZC2AF0C2EE49BC591/  NATS 137
     2      ,IFCN/1/,FCN/3HDEI,6HDEECNE,6HDEXPEI/             NATS 138
C     ----------                                              NATS 139
C                                                             NATS 140
C     ZERO OF EI(X)                                           NATS 141
C                                                             NATS 142
C     ----------                                              NATS 143
      DATA X0/Z405F5CA54AD2D7F1/,X01/Z405F5CA54AD20000/,      NATS 144      5
     X        X02/Z36D7F0F264C30000/                          NATS 145
C     ----------                                              NATS 146
C                                                             NATS 147
C     COEFFICIENTS FOR R(5,5) APPROXIMATION,                  NATS 148
C     USED FOR -1.0 .LE. X .LT. 0.0                           NATS 149
C                                                             NATS 150
C     ----------                                              NATS 151
      DATA  A/ZC093C467E37DFGC8,Z40C110E996FE3178,Z402130CCA0D570ED,  NATS 152   6
     X        Z3F625544F457AA6,Z3E569011F23A2CC8,Z3D44F80EBCEB91CF/   NATS 153
      DATA  B/Z4110000000000000,Z406C07BAC4D04469,Z40146C6D6F730856,  NATS 154   7
     X        Z3F220150E697CE8E,Z3E1FF0F4A6DA9AF7,Z3CDB33FAB422E1DA/  NATS 155
C     ----------                                              NATS 156
C                                                             NATS 157
C     COEFFICIENTS FOR R(7,7) APPROXIMATION,                  NATS 158
C     USED FOR -4.0 .LE. X .LT. -1.0                          NATS 159
C                                                             NATS 160
C     ----------                                              NATS 161
      DATA  C/Z3D174D1AD3E2E266,Z40FFFFD4D38BC730,Z41DD92AE16CB369D,  NATS 162   8
     X        Z422C97D311F74081,Z4245FD8DCPDDF7F2,Z422A352C0E187B8C,  NATS 163
     X        Z41C6D33281818C45,Z4066C0AFC43F717/                     NATS 164
      DATA  D/Z411000000000000,Z41CD9233632006CC,Z4238717FD76B3FA3,   NATS 165   9
     X        Z426AA52AC37780A8,Z4259BB2A31911955,Z421F7F4782BEC4CE,  NATS 166
     X        Z413C3ABC9E69B478,Z401743F10E4FE979/                    NATS 167
C     ----------                                              NATS 168
C                                                             NATS 169
C     COEFFICIENTS FOR R(7,7) APPROXIMATION,                  NATS 170
C     USED FOR X .LT. -4.0                                    NATS 171
C                                                             NATS 172
C     ----------                                              NATS 173
      DATA  E/ZC0FFFFFFFFFFF884,ZC22267FCDOC105CF,ZC31AB885D23000FD,  NATS 174   10
```

```
      X        ZC395C04F996D432F,ZC4181BCA21430F9D,ZC419B018D42BE7A2,    NATS 179
      X        ZC383A13CE9F21FEA,ZC1EF62A6E33B49AD/                      NATS [illegible]
       DATA  F/Z4110000000000000,Z422467FCB0C2E00F,Z431EE585685725C3,    NATS [illegible]   11
      X        Z43C7645BA5C7422F,Z442B6113476C61CF,Z443FC4253609BA4D,    NATS [illegible]
      X        Z442B8DC679374D02,Z4394A2394EA64382/                      NATS [illegible]
C      ----------                                                        NATS [illegible]
C                                                                        NATS [illegible]
C      COEFFICIENTS FOR R(5,5) APPROXIMATION FOR                         NATS 182
C      LN(X/X0),  ABS(1-X/X0) .LT. .1                                    NATS 193
C                                                                        NATS 184
C      ----------                                                        NATS [illegible]
       DATA P0/Z429889F12723EF63,Z4315444AE68C49FC,Z43103BD17D0451EF,    NATS [illegible]   12
      X        Z424CDEF79E7D4F0A,Z41775E404E1F69C8,Z3FE4572849F33E48/    NATS [illegible]
       DATA Q0/Z429889F12723EF63,Z431A039A6FC56977,Z431A1294596B9C59,    NATS [illegible]   13
      X        Z4209BC89B4DDA489,Z4222F75389A8CCCD,Z4120000000000000/    NATS [illegible]
C      ----------                                                        NATS 190
C                                                                        NATS 191
C      COEFFICIENTS FOR R(8,8) APPROXIMATION,                            NATS 192
C      IN CHEBYSHEV POLYNOMIAL FORM, USED FOR                            NATS 193
C      0.0 .LT. X .LT. 6.0                                               NATS 194
C                                                                        NATS [illegible]
C      ----------                                                        NATS 196
       DATA P1/Z415882FAB1C5ADD4,Z42CE5036A1E5C68A,Z4437C0581653B38E,    NATS [illegible]   14
      X        Z448FF963D6E088AD,Z46449068754DD3A4,ZC619B411D22D93E0,    NATS [illegible]
      X        Z481509C124CD86FB,ZC777652F57173F62,Z46D2AF8E86763BDA/    NATS [illegible]
       DATA Q1/Z4219A10000000000,ZC35E89E5032216A0,Z44A6C08CCD646C5F,    NATS [illegible]   15
      X        ZC5B6965493822757,Z468729386689CA00,ZC74521D75324E9D5,    NATS 201
      X        Z4817ED17C37935DF,ZC8535962239D6B97,Z484C45A46809BFBA/    NATS 202
C      ----------                                                        NATS 203
C                                                                        NATS 204
C      COEFFICIENTS FOR R(8,8) APPROXIMATION,                            NATS 205
C      IN J-FRACTION FORM, USED FOR                                      NATS 206
C      6.0 .LE. X .LT. 12.0                                              NATS 207
C                                                                        NATS [illegible]
C      ----------                                                        NATS [illegible]
       DATA P2/ZC[illegible]2782D3E7F8723F,ZC224C73F6[illegible]3A5C,Z421745E4D[illegible]E91A,    NATS [illegible]   16
      X        Z417E3[illegible]C81[illegible]34A2D,ZC213[illegible]5CDD0[illegible]F8D4,Z[illegible]5D2F710D3FF75,    NATS 211
      X        Z4142E579A88DBC12,Z415DB2DCJA7A3ACC,Z40FF8B90BDC62D00/    NATS 212
       DATA Q2/Z412A3CBE7[illegible]74E731,Z433C567DC545[illegible]873,ZC18433E6573[illegible]CF9,    NATS 213   17
      X        Z4313D478C9897198,Z423451C9DF0AD122,Z431555D7E91[illegible]485,    NATS [illegible]
      X        ZC2C7264C3364A115,Z4112570CE54009ED/                      NATS [illegible]
C      ----------                                                        NATS [illegible]
C                                                                        NATS [illegible]
C      COEFFICIENTS FOR R(9,9) APPROXIMATION,                            NATS [illegible]
C      IN J-FRACTION FORM, USED FOR                                      NATS [illegible]
C      12.0 .LE. X .LT. 24.0                                             NATS [illegible]
C                                                                        NATS 221
C      ----------                                                        NATS 222
       DATA P3/ZC11A5D10E04A8481,ZC21299D5F90F9F7F,ZC1A01A4AF4FE0826,    NATS [illegible]   18
      X        ZC215[illegible]F24A571E03,ZC0E9CA0F2A953185,ZC2213C72[illegible]090F2,    NATS 224
      X        Z4218F47206C8760A,Z421A869816A026F0,ZC11D857[illegible]B9F4,    NATS 225
      X        Z40FFFF8FC5509274/                                        NATS 226
       DATA Q3/Z[illegible]261EC8D9F6FP5A1,Z424009BAA237006F,Z4239F1069954BF52,    NATS 227   19
      X        Z42F[illegible]1CF4454[illegible],[illegible]C484C18765403,Z434A8D519ECB98C4,    NATS 228
      X        Z42C7196E[illegible]3321[illegible],[illegible]EFC[illegible]F363E13,Z411006485C45F248/    NATS 229
C      ----------                                                        NATS 230
C                                                                        NATS 231
C      COEFFICIENTS FOR R(9.9) APPROXIMATION,                            NATS 232
```

```
                                                                                    ISN
C      IN J-FRACTION FORM, USED FOR X .GE. 24.0                          NATS 233
C                                                                        NATS 234
C      ----------                                                        NATS 235
       DATA P4/Z42AF56B5AF030517,ZC2DF2JAF08369607,ZC21231F952F731CB,    NATS 236    20
      X         ZC21BFAD7A31F07FA,ZC17A1A87A4CC3F96,ZC1F49212D2DC262F,   NATS 237
      X         ZC17116FA4792119F,ZC1500045A1392820,ZC130000000DC93A1,   NATS 238
      X         Z41100000CC00C032/                                        NATS 239
       DATA Q4/Z449B689903F6E0E3,Z413F90786B65B672,Z4289CA5703B59D8A,    NATS 240    21
      X         Z42752DE1651338F6,Z42467BB1FE61EB5D,ZC1C04CE86F570E7F,   NATS 241
      X         ZC17FE10486DF7CDC,ZC12FFFFEE39110D8,Z411FFFFFFFF5B8A/    NATS 242
C                                                                        NATS 243
C      ------------------------------------------------------------      NATS 244
C                                                                        NATS 245
       X = ARG                                                           NATS 246    22
       JK = 0                                                            NATS 247    23
       GO TO (100,450,100), INT                                          NATS 248    24
   100 IF (X) 280, 640, 110                                              NATS 249    25
   110 IF (X .GE. 12.D0) GO TO 200                                       NATS 250    26
       IF (X .GE. 6.D0) GO TO 160                                        NATS 251    27
C      :::::::::: 0.0 .LE. X .LT. 6.0.                                   NATS 252
C                 RATIONAL APPROXIMATION USED IS EXPRESSED               NATS 253
C                 IN TERMS OF CHEBYSHEV POLYNOMIALS TO                   NATS 254
C                 IMPROVE CONDITIONING  ::::::::::                       NATS 255
       T = X + X                                                         NATS 256    28
       T = T / 3.0D0 - 2.0D0                                             NATS 257    29
       PX(1) = 0.0D0                                                     NATS 258    30
       QX(1) = 0.0D0                                                     NATS 259    31
       PX(2) = P1(1)                                                     NATS 260    32
       QX(2) = Q1(1)                                                     NATS 261    33
C                                                                        NATS 262
       DO 120 I = 2, 8                                                   NATS 263    34
          PX(I+1) = T * PX(I) - PX(I-1) + P1(I)                          NATS 264    35
          QX(I+1) = T * QX(I) - QX(I-1) + Q1(I)                          NATS 265    36
   120 CONTINUE                                                          NATS 266    37
C                                                                        NATS 267
       SUMP = .5D0 * T * PX(9) - PX(8) + P1(9)                           NATS 268    38
       SUMQ = .5D0 * T * QX(9) - QX(8) + Q1(9)                           NATS 269    39
       FRAC = SUMP / SUMQ                                                NATS 270    40
       XMX0 = (X - X01) - X02                                            NATS 271    41
       IF (DABS(XMX0) .LT. 0.037D0) GO TO 140                            NATS 272    42
       XX0 = X / X0                                                      NATS 273    43
       EI = DLOG(XX0) + XMX0 * FRAC                                      NATS 274    44
       IF (INT .EQ. 3) EI = DEXP(-X) * EI                                NATS 275    45
       GO TO 410                                                         NATS 276    46
C      :::::::::: EVALUATE APPROXIMATION FOR LN(X/X0)                    NATS 277
C                 FOR X CLOSE TO X0 ::::::::::                           NATS 278
   140 Y = XMX0 / X0                                                     NATS 279    47
       SUMP = ((((P0(6) * Y + P0(5)) * Y + P0(4)) * Y + P0(3)) * Y       NATS 280    48
      X          + P0(2)) * Y + P0(1)                                    NATS 281
       SUMQ = ((((Q0(6) * Y + Q0(5)) * Y + Q0(4)) * Y + Q0(3)) * Y       NATS 282    49
      X          + Q0(2)) * Y + Q0(1)                                    NATS 283
       EI = (SUMP / (SUMQ * X0) + FRAC) * XMX0                           NATS 284    50
       IF (INT .EQ. 3) EI = DEXP(-X) * EI                                NATS 285    51
       GO TO 410                                                         NATS 286    52
C      :::::::::: 6.0 .LE. X .LT. 12.0 ::::::::::                        NATS 287
   160 FRAC = 0.0D0                                                      NATS 288    53
C                                                                        NATS 289
       DO 180 I = 1, 8                                                   NATS 290    54
```

```
                                                                   ISN
      160 FRAC = Q2(I) / (P2(I) + X + FRAC)              NATS 291    55
C                                                        NATS 292
          EI = (P2(9) + FRAC) / X                        NATS 293    56
          IF (INT .EQ. 3) GO TO 410                      NATS 294    57
          EI = EI * DEXP(X)                              NATS 295    58
          GO TO 410                                      NATS 296    59
C         :::::::::: 12.0 .LE. X .LT. 24.0 ::::::::::    NATS 297
      200 IF (X .GE. 24.D0) GO TO 240                    NATS 298    60
          FRAC = 0.0D0                                   NATS 299    61
C                                                        NATS 300
          DO 220 I = 1, 9                                NATS 301    62
      220 FRAC = Q3(I) / (P3(I) + X + FRAC)              NATS 302    63
C                                                        NATS 303
          EI = (P3(10) + FRAC) / X                       NATS 304    64
          IF (INT .EQ. 3) GO TO 410                      NATS 305    65
          EI = EI * DEXP(X)                              NATS 306    66
          GO TO 410                                      NATS 307    67
C         :::::::::: 24.0 .LE. X ::::::::::              NATS 308
      240 IF ((X .GE. XMAX) .AND. (INT .LT. 3)) GO TO 620 NATS 309   68
          Y = 1.0D0 / X                                  NATS 310    69
          FRAC = 0.0D0                                   NATS 311    70
C                                                        NATS 312
          DO 260 I = 1, 9                                NATS 313    71
      260 FRAC = Q4(I) / (P4(I) + X + FRAC)              NATS 314    72
C                                                        NATS 315
          FRAC = P4(10) + FRAC                           NATS 316    73
          EI = Y + Y * Y * FRAC                          NATS 317    74
          IF (INT .EQ. 3) GO TO 410                      NATS 318    75
          IF (X .GT. 170.0D0) GO TO 270                  NATS 319    76
          EI = EI * DEXP(X)                              NATS 320    77
          GO TO 410                                      NATS 321    78
C         :::::::::: CALCULATION REFORMULATED TO AVOID   NATS 322
C                    PREMATURE OVERFLOW ::::::::::       NATS 323
      270 EI = (EI * DEXP(X-40.0D0)) * DEXP40            NATS 324    79
          GO TO 410                                      NATS 325    80
C         :::::::::: ORIGINAL X WAS NEGATIVE.  CALCULATION NATS 326
C                    OF OPPOSITE JOINS AT LABEL 300 :::::::::: NATS 327
      280 Y = -X                                         NATS 328    81
      300 W = 1.0D0 / Y                                  NATS 329    82
          IF (Y .GT. 4.0D0) GO TO 340                    NATS 330    83
          IF (Y .GT. 1.0D0) GO TO 320                    NATS 331    84
C         :::::::::: 0.0 .LT. -X .LE. 1.0 ::::::::::     NATS 332
          EI = DLOG(Y) - (((((A(6) * Y + A(5)) * Y + A(4)) NATS 333  85
     1           * Y + A(3)) * Y + A(2)) * Y + A(1)) /   NATS 334
     2        ((((B(6) * Y + B(5)) * Y + B(4))           NATS 335
     3           * Y + B(3)) * Y + B(2)) * Y + B(1))     NATS 336
          IF (INT .EQ. 3) EI = EI*DEXP(Y)                NATS 337    86
          GO TO 400                                      NATS 338    87
C         :::::::::: 1.0 .LT. -X .LE. 4.0 ::::::::::     NATS 339
      320 EI = -((((((C(8) * W + C(7)) * W + C(6)) * W + C(5)) NATS 340 88
     1           * W + C(4)) * W + C(3)) * W + C(2)) * W + C(1)) / NATS 341
     2        ((((((D(8) * W + D(7)) * W + D(6)) * W + D(5)) NATS 342
     3           * W + D(4)) * W + D(3)) * W + D(2)) * W + D(1)) NATS 343
          IF  (INT .EQ. 3) GO TO 410                     NATS 344    89
          EI = EI*DEXP(-Y)                               NATS 345    90
          GO TO 400                                      NATS 346    91
C         :::::::::: 4.0 .LT. -X ::::::::::              NATS 347
      340 IF ((-EI*Y(X) .LT. XMIN) .AND. (INT .LT. 3)) GO TO 600 NATS 348 92
```

```
                                                                          ISN
      EI = -W * (1.0D0 + W * (((((((E(8) * W + E(7))       NATS 349       93
     1        * W + E(6)) * W + E(5)) * W + E(4))          NATS 350
     2         * W + E(3)) * W + E(2)) * W + E(1)) /       NATS 351
     3       (((((((F(8) * W + F(7)) * W + F(6)) * W + F(5))  NATS 352
     4         * W + F(4)) * W + F(3)) * W + F(2)) * W + F(1)))  NATS 353
      IF (INT .EQ. 3) GO TO 410                            NATS 354       94
      EI = EI * DEXP(-Y)                                   NATS 355       95
  400 IF (INT .EQ. 2) EI = -EI                             NATS 356       96
      IF (JK .EQ. 4) GO TO 680                             NATS 357       97
  410 RESULT = EI                                          NATS 358       98
      RETURN                                               NATS 359       99
  450 Y = X                                                NATS 360      100
      IF (Y) 660,640,300                                   NATS 361      101
C     :::::::::: ERROR RETURN FOR X .LT. XMIN,             NATS 362
C                CAUSING UNDERFLOW ::::::::::              NATS 363
  600 EI = 0.0D0                                           NATS 364      102
      JK = 1                                               NATS 365      103
      GO TO 680                                            NATS 366      104
C     :::::::::: ERROR RETURN FOR X .GT. XMAX,             NATS 367
C                CAUSING OVERFLOW ::::::::::               NATS 368
  620 EI = XINF                                            NATS 369      105
      JK = 2                                               NATS 370      106
      GO TO 680                                            NATS 371      107
C     :::::::::: ERROR RETURN FOR ILLEGAL                  NATS 372
C                ARGUMENT, X = 0 ::::::::::                NATS 373
  640 EI = -XINF                                           NATS 374      108
      IF (INT .EQ. 2) EI = -EI                             NATS 375      109
      JK = 3                                               NATS 376      110
      GO TO 680                                            NATS 377      111
C     :::::::::: SET UP ERROR RETURN FOR ARGUMENT          NATS 378
C                .LT. 0 IN DEXONE ::::::::::               NATS 379
  660 JK = 4                                               NATS 380      112
      GO TO 280                                            NATS 381      113
C     :::::::::: UPDATE ERROR COUNTS, ETC. ::::::::::      NATS 382
  680 CALL FCNMON(IFCN,JK,FCN(INT),ARG,EI)                 NATS 383      114
      GO TO 410                                            NATS 384      115
C     :::::::::: LAST CARD OF NATSEI ::::::::::            NATS 385
      END                                                  NATS 386      116
```

```
                                                                        ISN
C NAME [illegible] DEI        FTN    06-24-75     THE UNIV OF MICH COMP CTR  DEI   1
C                                                                DEI   2
C      ------- FUNPACK ------- IBM S/360 ------ LONG PRECISION --------- DEI   3
C                                                                DEI   4
       FUNCTION DEI(X)                                           DEI   5
C                                                                DEI   6
       REAL*8 DEI,X,RESULT                                       DEI   7       2
       INTEGER INT                                               DEI   8       3
C                                                                DEI   9
C      THIS LONG PRECISION SUBROUTINE COMPUTES APPROXIMATE VALUES  DEI  10
C      FOR THE EXPONENTIAL INTEGRAL  EI(X), WHERE  X  IS REAL.   DEI  11
C                                                                DEI  12
C      OTHER  FUNPACK  SUBPROGRAMS REQUIRED -  NATSEI             DEI  13
C                                                                DEI  14
C      ------------------------------------------                DEI  15
C                                                                DEI  16
       INT = 1                                                   DEI  17       4
       CALL NATSEI(X,RESULT,INT)                                 DEI  18       5
       DEI = RESULT                                              DEI  19       6
       RETURN                                                    DEI  20       7
C      ---------- LAST CARD OF DEI ----------                    DEI  21
       END                                                       DEI  22       8
```

```
C NAASA 10.1.034 FCNMON   PTN   06-24-75     THE UNIV OF MICH COMP CTR  FCNM   1
C                                                                       FCNM   2
C     ------- FUNPACK ------- IBM S/360 ------ LONG PRECISION ---------  FCNM   3
C                                                                       FCNM   4
      SUBROUTINE FCNMON(IFCN,JERR,FCN,ARG,RESULT)                       FCNM   5        1
C                                                                       FCNM   6
      REAL*8 ARG,FCN,RESULT                                             FCNM   7        2
      INTEGER ERCONT(20),IERCNT(20,5),IFCN,J,JERR,JK,K,NERR(20),        FCNM   8        3
     1         NERINT                                                   FCNM   9
      LOGICAL IPRINT(20)                                                FCNM  10        4
C                                                                       FCNM  11
C     THIS ROUTINE IS INTENDED FOR USE ONLY BY OTHER ELEMENTS OF        FCNM  12
C     FUNPACK.  ALL ERROR DIAGNOSTIC FACILITIES AND PRINT STATEMENTS    FCNM  13
C     ARE CONCENTRATED WITHIN THIS SUBROUTINE.  CALLS BY INDIVIDUAL     FCNM  14
C     FUNPACK  FUNCTION PACKETS USE THE PARAMETERS                      FCNM  15
C                                                                       FCNM  16
C     IFCN     - AN INTEGER IDENTIFYING THE CALLING PACKET.             FCNM  17
C                                                                       FCNM  18
C     JERR     - AN INTEGER IDENTIFYING THE ERROR DETECTED WITHIN THE   FCNM  19
C                CALLING PACKET.                                        FCNM  20
C                                                                       FCNM  21
C     FCN      - A 6 CHARACTER HOLLERITH STRING IDENTIFYING THE ACTIVE  FCNM  22
C                ENTRY IN THE CALLING PACKET.                           FCNM  23
C                                                                       FCNM  24
C     ARG      - THE REAL*8 ARGUMENT LEADING TO THE ERROR CONDITION.    FCNM  25
C                                                                       FCNM  26
C     RESULT   - THE REAL*8 FUNCTION VALUE BEING RETURNED BY THE        FCNM  27
C                CALLING PACKET.                                        FCNM  28
C                                                                       FCNM  29
C     CALLS TO THIS ROUTINE FROM THE ERROR MONITORING SUBPROGRAM        FCNM  30
C     MONERR  SPECIFY ONLY THE PARAMETERS                               FCNM  31
C                                                                       FCNM  32
C     IFCN     - THE PARAMETER OF THE SAME NAME INPUT TO  MONERR,       FCNM  33
C                                                                       FCNM  34
C     JERR     - THE VALUE  -KK-2, WHERE  KK  IS AN INPUT PARAMETER TO  FCNM  35
C                MONERR,                                                FCNM  36
C                                                                       FCNM  37
C     ALL OTHER PARAMETERS BEING DUMMIES.                               FCNM  38
C                                                                       FCNM  39
C     ERROR INFORMATION IS STORED IN  FCNMON  IN ARRAYS WITH THE FIRST  FCNM  40
C     SUBSCRIPT KEYED TO  IFCN.  THESE ARRAYS ARE DESCRIBED BELOW.      FCNM  41
C                                                                       FCNM  42
C     IPRINT    - A LOGICAL ARRAY AUTHORIZING THE PRINTING OF ERROR     FCNM  43
C                 MESSAGES.                                             FCNM  44
C                                                                       FCNM  45
C     ERCONT    - AN INTEGER ARRAY AUTHORIZING THE TERMINATION (VALUE   FCNM  46
C                 OF -1) OR CONTINUATION (ANY OTHER VALUE) OF THE       FCNM  47
C                 COMPUTER RUN AFTER ERROR DETECTION.                   FCNM  48
C                                                                       FCNM  49
C     IERCNT    - AN INTEGER ARRAY TABULATING THE FREQUENCY OF THE      FCNM  50
C                 VARIOUS ERRORS REPORTED BY EACH FUNCTION PACKET.      FCNM  51
C                                                                       FCNM  52
C     NERR      - AN INTEGER ARRAY DESIGNATING THE LAST ERROR REPORTED  FCNM  53
C                 BY EACH FUNCTION PACKET.                              FCNM  54
C                                                                       FCNM  55
C     ----------                                                        FCNM  56
C                                                                       FCNM  57
C     QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW         FCNM  58
```

```
C     APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY
C
C     ---------------------------------------------------------------
C
C
      DATA NPRINT/6/,IPRINT/20*.TRUE./,ERCONT/20*0/,IERCNT/100*0/,          5
     1     NERR/20*0/
C
C     ---------------------------------------------------------------
C
      IF (JERR .LT. 0) GC TC 500                                            6
C     ---------- ERROR MESSAGE FROM FUNCTION ROUTINE ----------
      IERCNT(IFCN,JERR) = IERCNT(IFCN,JERR) + 1                             7
      IF (IPRINT(IFCN)) WRITE(NPRINT,1001) FCN,ARG                          8
      NERR(IFCN) = JERR                                                     9
      IF (ERCONT(IFCN) .EQ. -1) GO TO 200                                   10
      IF (IPRINT(IFCN)) WRITE(NPRINT,1002) RESULT                           11
      IF (ERCONT(IFCN) .LT. 1) IPRINT(IFCN) = .FALSE.                       12
      GC TC 600                                                             13
C     ---------- PREPARE FOR TERMINATION AND TRACEBACK ----------
  200 DO 210 I = 1, 5                                                       14
        K = 6 - I                                                           15
        IF (IERCNT(IFCN,K) .NE. 0) GO TO 220                                16
  210 CONTINUE                                                              17
C     --------- TERMINATE AND TRACEBACK ----------
  220 WRITE(NPRINT,1003) (J,IERCNT(IFCN,J), J=1,K)                          18
      CALL ERRTRA                                                           19
      STOP                                                                  20
C     ---------- USER INTERROGATION AND RESETTING OF FLAGS ----------
  500 JK = -JERR - 2                                                        21
      IF (JK .GT. 3) GC TC 510                                              22
      J = JK + 2                                                            23
      GC TO (530,520,530,550,540), J                                        24
  510 JK = IERCNT(IFCN,JK-3)                                                25
      GO TC 580                                                             26
C
  520 DO 525 K = 1, 5                                                       27
  525   IERCNT(IFCN,K) = 0                                                  28
C
  530 IPRINT(IFCN) = .TRUE.                                                 29
      GC TO 570                                                             30
  540 JK = NERR(IFCN)                                                       31
      NERR(IFCN) = 0                                                        32
      GO TC 580                                                             33
  550 IPRINT(IFCN) = .FALSE.                                                34
  570 ERCONT(IFCN) = JK                                                     35
  580 JERR = JK                                                             36
  600 RETURN                                                                37
 1001 FORMAT(21H0ILLEGAL ARGUMENT IN ,A6,8H, ARG = ,D24.16)                 38
 1002 FORMAT(6X,35HEXECUTION CONTINUING WITH RESULT = ,D24.16)              39
 1003 FORMAT(6X,43HEXECUTION TERMINATING. ERROR COUNTS FOLLOW,/             40
     1  (6X,10HERROR NO. ,I2,3H = ,I6))
C     ---------- LAST CARD OF FCNMON ----------
      END                                                                   41
```

ISN

```
C
C
C*******************************
C
C   SUBROUTINE ERRTRA CALLED BY FCNMCN
      SUBROUTINE ERRTRA                    1
      PRINT 1                              2
    1 FORMAT(' ***ERRTRA CALLED****')      3
      CALL SYSTEM                          4
      RETURN                               5
      END                                  6
```

```
                                                              ISN
C
C***********************************
C
      SUBROUTINE NORM(X1,Y1,X2,Y2,AN1,AN2)                      1
C
C         GIVEN TWO POINTS, NORM CONSTRUCTS A LINE AND THE NORMAL TO IT. THE
C         NORMAL COMPONENTS ARE AN1 AND AN2.
C
      IMPLICIT REAL*8 (A-H,O-Z)                                 2
      B=DSQRT((X2-X1)*(X2-X1)+(Y2-Y1)*(Y2-Y1))                  3
      IF(B .LE. 1.D-7) GO TO 10                                 4
      AN1=(Y1-Y2)/B                                             5
      AN2=(X2-X1)/B                                             6
      RETURN                                                    7
   10 WRITE(6,20) X1,Y1                                         8
      STOP 1                                                    9
   20 FORMAT(///' ***ERROR***'//5X,'DUPLICATE HULL POINTS AT',  10
     1 2(F10.4))
      END                                                      11
```

```
C
C**********************************
C
      SUBROUTINE HULLP(X1,Y1,X2,Y2,X3,Y3,ICHINE,N1,N2,CURV,ARC1,ARC2)            1
C
C        HULLP TAKES THREE POINTS, PT1, PT2, PT3, AND RETURNS THE NORMAL,
C        N1, N2, THE CURVATURE, CURV, AND HALF THE ARC LENGTH TO EITHER SIDE
C        OF THE MIDDLE POINT, ARC1, ARC2.
C
      IMPLICIT REAL*8 (A-H,O-Z)                                                  2
      REAL*8 N1,N2,N1P,N2P                                                       3
C
C        IF ICHINE .GT. 0, THE MIDDLE POINT, PT2, IS DESIGNATED AS A CHINE.
C
      IF(ICHINE .EQ. 0) GO TO 10                                                 4
C
C        TWO   LINES    ARE DRAWN BETWEEN PT1,PT2 AND PT2,PT3. THE NORMAL AT
C        PT2 IS THE AVERAGE OF THE NORMALS TO THE TWO LINES.
C
      CURV=0.D0                                                                  5
      CALL NORM(X1,Y1,X2,Y2,AN11,AN12)                                           6
      CALL NORM(X2,Y2,X3,Y3,AN21,AN22)                                           7
      RN=DSQRT(2.D0*(1.D0+AN11*AN21+AN12*AN22))                                  8
      N1=(AN11+AN21)/RN                                                          9
      N2=(AN12+AN22)/RN                                                         10
      ARC1=DSQRT((X1-X2)*(X1-X2)+(Y1-Y2)*(Y1-Y2))*0.5D0                         11
      ARC2=DSQRT((X2-X3)*(X2-X3)+(Y2-Y3)*(Y2-Y3))*0.5D0                         12
      RETURN                                                                    13
   10 A=X1*(Y2-Y3)-Y1*(X2-X3)+(X2*Y3-X3*Y2)                                     14
C
C        CHECK TO SEE IF THE THREE POINTS LIE ON A STRAIGHT LINE, BEFORE
C        A CIRCLE IS FITTED. THE EQUATION FOR THE CIRCLE COMES FROM THOMAS,
C        CALCULUS AND ANALYTIC GEOMETRY, PAGE 463.
C
      IF(DABS(A) .GE. 1.D-5) GO TO 20                                           15
      CALL NORM(X1,Y1,X3,Y3,N1,N2)                                              16
      CURV=0.D0                                                                 17
      ARC1=DSQRT((X1-X2)*(X1-X2)+(Y1-Y2)*(Y1-Y2))*0.5D0                         18
      ARC2=DSQRT((X2-X3)*(X2-X3)+(Y2-Y3)*(Y2-Y3))*0.5D0                         19
      RETURN                                                                    20
   20 SQ1=X1*X1+Y1*Y1                                                           21
      SQ2=X2*X2+Y2*Y2                                                           22
      SQ3=X3*X3+Y3*Y3                                                           23
      D=-(SQ1*(Y2-Y3)-Y1*(SQ2-SQ3)+(Y3*SQ2-Y2*SQ3))                             24
      E=SQ1*(X2-X3)-X1*(SQ2-SQ3)+(SQ2*X3-SQ3*X2)                                25
      F=-(SQ1*(X2*Y3-X3*Y2)-X1*(SQ2*Y3-SQ3*Y2)+Y1*(SQ2*X3-SQ3*X2))              26
C
C        THE CENTER OF THE CIRCLE IS GIVEN AT (X0,Y0) AND THE RADIUS IS R.
C
      X0=-D/(2.D0*A)                                                            27
      Y0=-E/(2.D0*A)                                                            28
      R=DSQRT(X0*X0+Y0*Y0-F/A)                                                  29
      CALL NORM(X1,Y1,X2,Y2,N1P,N2P)                                            30
      SGN=1.D0                                                                  31
C
C        THE DOT PRODUCT BETWEEN THE VECTOR FROM THE CENTER OF THE CIRCLE
C        TO THE MID POINT OF A LINE BETWEEN PT1 AND PT2, AND THE OUTWARD
C        NORMAL AT THAT MID POINT TELLS WHETHER THE CURVE IS CONCAVE OR
```

```
                                                                    ISN
C          CONVEX, AND HENCE GIVES THE SIGN ON THE CURVATURE AND NORMAL
C          COMPONENTS.
C
      DOTP=((X1+X2)/2.D0-X0)*N1P+((Y1+Y2)/2.D0-Y0)*N2P               32
      IF(DOTP .LE. 0.D0) SGN=-1.D0                                    33
      N1=SGN*(X2-X0)/R                                                34
      N2=SGN*(Y2-Y0)/R                                                35
      CURV=SGN/R                                                      36
C
C          THE ANGLE OF THE CIRCULAR SEGMENT IS FOUND AND IF IT IS GREATER
C          THAT PI AN ERROR IS RETURNED.
C
      R1=DSQRT((X1-X0)*(X1-X0)+(Y1-Y0)*(Y1-Y0))                       37
      R2=DSQRT((X2-X0)*(X2-X0)+(Y2-Y0)*(Y2-Y0))                       38
      R3=DSQRT((X3-X0)*(X3-X0)+(Y3-Y0)*(Y3-Y0))                       39
C
C          (VI1,VI2) IS THE VECTOR FROM THE I-TH POINT TO THE ORGIN.
C
      V11=(X1-X0)/R1                                                  40
      V12=(Y1-Y0)/R1                                                  41
      V21=(X2-X0)/R2                                                  42
      V22=(Y2-Y0)/R2                                                  43
      V31=(X3-X0)/R3                                                  44
      V32=(Y3-Y0)/R3                                                  45
C
C          THET, THE1, THE2 ARE THE ANGLES FORMED BY THE SEGMENTS PT1-PT3,
C          PT1-PT2, AND PT2-PT3 RESPECTIVELY. NOTE THAT DARCOS RETURNS
C          VALUES FROM 0 TO PI.
C
      THET=DARCOS(V11*V31+V12*V32)                                    46
      THE1=DARCOS(V11*V21+V12*V22)                                    47
      THE2=DARCOS(V21*V31+V22*V32)                                    48
      T=DABS(THET-THE1-THE2)                                          49
      IF(T .GE. 1.D-4) GO TO 30                                       50
C
C          HALF THE ARC LENGTH BETWEEN PT1 AND PT2 IS ARC1 AND HALF THE ARC
C          LENGTH BETWEEN PT2 AND PT3 IS ARC2.
C
      ARC1=THE1*R*0.5D0                                               51
      ARC2=THE2*R*0.5D0                                               52
      RETURN                                                          53
   30 WRITE(6,50) X2,Y2                                               54
      STOP 2                                                          55
   50 FORMAT(///5X,'***ERROR***'///10X,'THE POINTS AROUND ',F6.4,     56
     1 F6.4,' GIVE AN ARC OF MORE THAN PI*R')
      END                                                             57
```

```
C
C**********************************
C
      SUBROUTINE GRFUN(POTRE,POTIM,PNRE,PNIM)
C
C         GRFUN EVALUATES G(P,Q) AND DG(P,Q)/DN, WHERE G(P,) IS THE
C         COMPLEX SOLUTION OF A SOURCE IN A FLUID DOMAIN THAT SATISFIES
C         HELMHOLTZ'S EQUATION AND A LINEAR FREE SURFACE BOUNDARY CONDITION.
C         G IS COMPOSED OF A KO BESSEL FUNCTION PLUS A FREE SURFACE INTEGRAL.
C         IF P=Q, THE SINGULAR NATURE OF KO(R) REQUIRES THAT IT BE LEFT
C         OFF. HENCE, ONLY THE IMAGE, KO(RP), AND THE FREE SURFACE INTEGRAL
C         ARE RETURNED. IF RP=0. (P,Q ON THE FREE SURFACE) THEN KO(RP) IS
C         ALSO LEFT OFF. THE TOTAL VALUE OF DG/DN IS RETURNED SINCE IT IS NOT
C         SINGULAR AT P=Q. POTRE AND POTIM ARE THE REAL AND IMAGINARY
C         PART OF THE POTENTIAL WHILE PNRE AND PNIM ARE THE REAL AND IMAGINARY
C         PART OF DG/DN.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /TEST/ X,XP,Y,YP,K,BK(12),ICOUNT
      COMMON /DATA/ CURV,N1,N2
       EXTERNAL FNT1,FNT2,FNT3
      REAL*8 K,N1,N2
      DATA PI/3.14159265300/
      POTRE=0.D0
      POTIM=0.D0
      IF(K .LE. 1.D-6)   K=1.D-6
      IF(K .GE. 0.9999D0) K=0.9999D0
      X1=DABS(X-XP)
      R=DSQRT((X-XP)*(X-XP)+(Y-YP)*(Y-YP))
      RP=DSQRT((X-XP)*(X-XP)+(Y+YP)*(Y+YP))
      ARG=K*R
       ARGP=K*RP
      SQR=DSQRT(1.D0-K*K)
C
C      FIND THE K BESSEL FUNCTION  TERM OF THE POTENTIALS
C
      IF( RP  .LE. 1.D-5) GO TO 5
      CALL BESK(ARGP,0,BK0P,IER)
      CALL BESK(ARGP,1,BK1P,IER)
      IF( R  .LE. 1.D-5) GO TO 3
      CALL BESK(ARG,1,BK1,IER)
      PNRE=-K*(N1*(X-XP)*(BK1/R+BK1P/RP)+
     1        N2*[illegible]BK1/R+(Y+YP)*BK1P/RP))
      GO TO 9
    3 PNRE=-CURV/2.D0 K*N[illegible]*BK1P
      GO TO 9
    5 PNRE=-CURV
      PNIM=0.D0
      POTRE=-2.D0*DLOG(1.D0/K+DSQRT(1.D0/(K*K)-1.D0))/SQR
      POTIM=2.D0*PI/SQR
      RETURN
C
C         THE FREE SURFACE INTEGRAL IS EVALUATED EITHER THROUGH NUMERICAL
C         INTEGRATION OR BY A SERIES EXPANSION.
C
    9 PNIM=0.D0
      POTRE=0.D0
      [illegible]-Y-YP)
```

```
      Y2=Y+YP
C
C     CHECK TO SEE IF THE SINGULARITY IS TO BE SUBTRACTED
C     FROM THE INTEGRAND
C
      IF(X1 .GE. 2.5D0) GO TO 100
C
C      MODIFIED INTEGRAND OR SERIES REPRESENTATION
C
      IF(K .LT. 1.D-4) GO TO 20
      IF(K .LT. .1D0) GO TO 11
      IF(K .LT. .25D0) GO TO 12
      IF(K .LT. .5D0) GO TO 13
      GO TO 14
   11 IF (X1 .GE. 1.5D0 .OR. Y2 .GE. 5.D0) GO TO 20
      IF(X1 .GE. 1.D0 .OR. Y2 .GE. 3.D0) GO TO 30
      GO TO 40
   12 IF(Y2 .GE. 7.D0) GO TO 20
      IF(X1 .GE. 1.5D0) GO TO 30
      GO TO 50
   13 IF(Y2 .GE. 9.D0) GO TO 20
      IF(X1 .GE. 2.D0) GO TO 30
      GO TO 52
   14 IF(Y2 .GE. 10.D0) GO TO 20
      IF(Y2 .GE. 9.D0) GO TO 30
      GO TO 54
C
C      INTEGRATION OF MODIFIED INTEGRAND WITH A 4 POINT QUADRATURE
C
   20 ICOUNT=0
      CALL DQL4(FNT1,VAL1)
      ICOUNT=0
      CALL DQL4(FNT3,VAL2)
      GO TO 70
C
C      INTEGRATION OF MODIFIED INTEGRAND WITH A 8 POINT QUADRATURE
C
   30 ICOUNT=0
      CALL DQL8(FNT1,VAL1)
      ICOUNT=0
      CALL DQL8(FNT3,VAL2)
      GO TO 70
C
C      INTEGRATION OF MODIFIED INTEGRAND WITH A 12 POINT QUADRATURE
C
   40 ICOUNT=0
      CALL DQL12(FNT1,VAL1)
      ICOUNT=0
      CALL DQL12(FNT3,VAL2)
      GO TO 70
C
C      SET THE ERROR LIMIT FOR SXY
C
   50 EPS=1.D-50
      GO TO 60
   52 EPS=1.D-40
      GO TO 60
   54 EPS=1.D-30
```

```
      CONTINUE                                                       74
C
C   SERIES REPRESENTATION FOR SMALL KR
C
   60 X2=X-XP                                                        75
      SI=DSIN(X2*SCR)                                                76
      CS=DCOS(X2*SCR)                                                77
      ZETA=DATAN2(X2,Y2)                                             78
      AL=DLOG(1.D0/K*DSQRT(1.D0/(K*K) -1.D0));                       79
      CAL=1.D0/SCR                                                   80
      CALL SXY(X2,Y2,K,S2,S2X,S2Y,EPS)                               81
      PNRE=PNRE+N1*(AL*2.D0*EX*SI+4.D0*CAL*S2X)                      82
     1          +N2*(AL*2.D0*EX*CS/SCR+4.D0*CAL*S2Y)
      PNIM=PNIM-PI*2.D0*EX*(N1*SI+N2*CS/SCR)                         83
      POTRE=BKOP-AL*2.D0*EX*CS/SCR+4.D0*CAL*S2                       84
      POTIM=2.D0*PI*EX*CS/SCR                                        85
      RETURN                                                         86
C
C    EXPONENTIAL INTEGRAL ADDITION TO MODIFIED INTEGRAL
C
   70 Y1=-Y-YP                                                       87
      IF(X1 .LE. 1.D-5) GO TO 90                                     88
      CALL DEICOM(Y1,X1,EREAL,EIMG,IER)                              89
      SGN=(X-XP)/X1                                                  90
      CS=DCOS(X1)                                                    91
      SI=DSIN(X1)                                                    92
      PNRE=PNRE+N1*(-SGN*2.D0*EX*(EREAL*SI+EIMG*CS))                 93
     1          +N2*(-2.D0*EX*(EREAL*CS-EIMG*SI))
      POTRE=2.D0*EX*(EREAL*CS-EIMG*SI)                               94
      GO TO 140                                                      95
C
C     X1=0 BUT Y1 .NE. 0
C
   90 CALL DEICOM(Y1,X1,EREAL,EIMG,IER)                              96
      PNRE=PNRE+N2*(-2.D0*K*VAL2-2.D0*EX*EREAL)                      97
      PNIM=PNIM-2.D0*PI*EX/SCR *N2                                   98
       POTRE=-BKOP+2.D0*K*VAL2+2.D0*EX*EREAL                         99
      POTIM=2.D0*PI*EX/SCR                                          100
      RETURN                                                        101
C
C     STANDARD INTEGRATION OF THE INTEGRAND
C
  100 IF(X1 .GE. 6.D0 .OR. Y2 .GE. 10.D0) GO TO 110                 102
      IF(X1 .GE. 3.D0 .OR. Y2 .GE. 5.D0) GO TO 120                  103
      GO TO 130                                                     104
C
C    INTEGRATION WITH A FOUR POINT QUADRATURE
C
  110 ICOUNT=0                                                      105
      CALL DQL4(FNT2,VAL1)                                          106
      ICOUNT=0                                                      107
      CALL DQL4(FNT3,VAL2)                                          108
      GO TO 140                                                     109
C
C    INTEGRATION WITH A 8 POINT QUADRATURE
C
  120 ICOUNT=0                                                      110
      CALL DQL8(FNT2,VAL1)                                          111
```

```
      ICOUNT=0
      CALL DCL8(FNT3,VAL2)
      GO TO 140
C
C      INTEGRATION WITH A 12 POINT QUADRATURE
C
  130 ICOUNT=0
      CALL DCL12(FNT2,VAL1)
      ICOUNT=0
      CALL DCL12(FNT3,VAL2)
  140 CS=DCOS((X-XI)*SQR)
      SI=DSIN((X-XI)*SQR)
      SGN=(X-XI)/X1
      PNRE=PNRE+N1*(2.D0*K*(X-XI)*VAL1-SGN*2.D0*PI*EX*CS)
     1         +N2*(-2.D0*K*VAL2+SGN*2.D0*PI*EX*SI/SQR)
      PNIM=PNIM-2.D0*PI*EX*(N1*SI+N2*CS/SQR)
      POTRE=-PKGP+2.D0*K*VAL2-2.D0*PI*EX*SGN*SI/SQR+POTRE
      POTIM=2.D0*PI*EX/SQR*CS
      RETURN
      END
```

```
C
C********************************
C
      SUBROUTINE BK1MOD(X,BK,IER)                                        1
C                                                                  BESK  10
C     .............................................................BESK  20
C                                                                  BESK  30
C        SUBROUTINE BK1MOD
C                                                                  BESK  50
C           COMPUTE THE K1 BESSEL FUNCTION MINUS ITS 1/X SINGULARITY
C            FOR A GIVEN ARGUMENT.
C                                                                  BESK  70
C        USAGE                                                     BESK  80
C           CALL BK1MOD(X,BK,IER)
C                                                                  BESK 100
C        DESCRIPTION OF PARAMETERS                                 BESK 110
C           X  -THE ARGUMENT OF THE K BESSEL FUNCTION DESIRED      BESK 120
C           BK -THE RESULTANT K BESSEL FUNCTION                    BESK 140
C           IER-RESULTANT ERROR CODE WHERE                         BESK 150
C              IER=0  NO ERROR                                     BESK 160
C              IER=2  X IS ZERO OR NEGATIVE                        BESK 180
C              IER=4  BK .GT. 10**70                               BESK 200
C                                                                  BESK 210
C                                                                  BESK 240
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED             BESK 250
C           NONE                                                   BESK 260
C                                                                  BESK 270
C        METHOD                                                    BESK 280
C           POLYNOMIAL APPROXIMATION TECHNIQUE
C           AS DESCRIBED BY A.J.M.HITCHCOCK,'POLYNOMIAL APPROXIMATIONS BESK 330
C           TO BESSEL FUNCTIONS OF ORDER ZERO AND ONE AND TO RELATED   BESK 340
C           FUNCTIONS', M.T.A.C., V.11,1957,PP.86-88, AND G.N. WATSON, BESK 350
C           'A TREATISE ON THE THEORY OF BESSEL FUNCTIONS', CAMBRIDGE  BESK 360
C           UNIVERSITY PRESS, 1958, P. 62                          BESK 370
C                                                                  BESK 380
C     .............................................................BESK 390
C                                                                  BESK 400
      IMPLICIT REAL*8 (A-H,O-Z)                                    BESK 410      2
      DIMENSION T(12)                                              BESK 420      3
      BK=.0                                                        BESK 430      4
      IF(X)12,12,20                                                              5
   12 IER=2                                                        BESK 480      6
      RETURN                                                       BESK 490      7
   20 IF(X-170.0)22,22,21                                          BESK 500      8
   21 IER=3                                                        BESK 510      9
      RETURN                                                       BESK 520     10
   22 IER=0                                                        BESK 530     11
      IF(X-1.)36,36,25                                             BESK 540     12
   25 A=DEXP(-X)                                                                13
      B=1./X                                                       BESK 560     14
      C=DSQRT(B)                                                                15
      T(1)=B                                                       BESK 580     16
      DO 26 L=2,12                                                 BESK 590     17
   26 T(L)=T(L-1)*B                                                BESK 600     18
C                                                                  BESK 720
C     COMPUTE K1 USING POLYNOMIAL APPROXIMATION                    BESK 730
C                                                                  BESK 740
      G1=A*(1.2533141+.46999270*T(1)-.14685830*T(2)+.12804266*T(3)) BESK 750    19
```

ISN

```
      2-.17364316*T(4)+.28476181*T(5)-.45943421*T(6)+.62933807*T(7)    BESK 760
      3-.66322954*T(8)+.50502386*T(9)-.25813038*T(10)+.078800012*T(11) BESK 770
      4-.010924177*T(12))*C                                            BESK 780
       BK=G1-1.D0/X
       RETURN
   36 B=X/2.                                                           BESK 810
      A=-.57721566+DLOG(B)                                             BESK 840
      C=B*B
C                                                                      BESK 960
C     COMPUTE K1 USING SERIES EXPANSION                                BESK1140
C                                                                      BESK1150
      X2J=B                                                            BESK1160
      FACT=1.
      HJ=1.                                                            BESK1180
      G1=X2J*(.5+A-HJ)                                                 BESK1190
      DO 50 J=2,6
      X2J=X2J*C                                                        BESK1210
      RJ=1./DFLOAT(J)                                                  BESK1220
      FACT=FACT*RJ*RJ
      HJ=HJ+RJ                                                         BESK1240
   50 G1=G1+X2J*FACT*(.5+(A-HJ)*DFLOAT(J))                             BESK1250
      BK=G1
      RETURN
      END                                                              BESK1290
                                                                       BESK1300
```

20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37

```
C
C***********************************
C
      SUBROUTINE BESK(X,N,BK,IER)                                       BESK  00
C                                                                       BESK  10
C     ..................................................................BESK  20
C                                                                       BESK  30
C        SUBROUTINE BESK                                                BESK  40
C                                                                       BESK  50
C           COMPUTE THE K BESSEL FUNCTION FOR A GIVEN ARGUMENT AND ORDERBESK  60
C                                                                       BESK  70
C        USAGE                                                          BESK  80
C           CALL BESK(X,N,BK,IER)                                       BESK  90
C                                                                       BESK 100
C        DESCRIPTION OF PARAMETERS                                      BESK 110
C           X  -THE ARGUMENT OF THE K BESSEL FUNCTION DESIRED           BESK 120
C           N  -THE ORDER OF THE K BESSEL FUNCTION DESIRED              BESK 130
C           BK -THE RESULTANT K BESSEL FUNCTION                         BESK 140
C           IER-RESULTANT ERROR CODE WHERE                              BESK 150
C              IER=0  NO ERROR                                          BESK 160
C              IER=1  N IS NEGATIVE                                     BESK 170
C              IER=2  X IS ZERO OR NEGATIVE                             BESK 180
C              IER=3  X .GT. 170, MACHINE RANGE EXCEEDED                BESK 190
C              IER=4  BK .GT. 10**70                                    BESK 200
C                                                                       BESK 210
C        REMARKS                                                        BESK 220
C           N MUST BE GREATER THAN OR EQUAL TO ZERO                     BESK 230
C                                                                       BESK 240
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                  BESK 250
C           NONE                                                        BESK 260
C                                                                       BESK 270
C        METHOD                                                         BESK 280
C           COMPUTES ZERO ORDER AND FIRST ORDER BESSEL FUNCTIONS USING  BESK 290
C           SERIES APPROXIMATIONS AND THEN COMPUTES N TH ORDER FUNCTION BESK 300
C           USING RECURRENCE RELATION.                                  BESK 310
C           RECURRENCE RELATION AND POLYNOMIAL APPROXIMATION TECHNIQUE  BESK 320
C           AS DESCRIBED BY A.J.M.HITCHCOCK,'POLYNOMIAL APPROXIMATIONS  BESK 330
C           TO BESSEL FUNCTIONS OF ORDER ZERO AND ONE AND TO RELATED    BESK 340
C           FUNCTIONS', M.T.A.C., V.11,1957,PP.86-88, AND G.N. WATSON,  BESK 350
C           'A TREATISE ON THE THEORY OF BESSEL FUNCTIONS', CAMBRIDGE   BESK 360
C           UNIVERSITY PRESS, 1958, P. 62                               BESK 370
C                                                                       BESK 380
C     ..................................................................BESK 390
C                                                                       BESK 400
      IMPLICIT REAL*8 (A-H,O-Z)                                         BESK 410
      DIMENSION T(12)                                                   BESK 420
      BK=.0                                                             BESK 430
      IF(N)10,11,11                                                     BESK 440
   10 IER=1                                                             BESK 450
      RETURN                                                            BESK 460
   11 IF(X)12,12,20                                                     BESK 470
   12 IER=2                                                             BESK 480
      RETURN                                                            BESK 490
   20 IF(X-170.0)22,22,21                                               BESK 500
   21 IER=3                                                             BESK 510
      RETURN                                                            BESK 520
   22 IER=0                                                             BESK 530
      IF(X-1.)36,36,25                                                  BESK 540
```

```
   25 A=DEXP(-X)
      B=1./X
      C=DSQRT(B)
      T(1)=B
      DO 26 L=2,12
   26 T(L)=T(L-1)*B
      IF(N-1)27,29,27
C
C     COMPUTE KO USING POLYNOMIAL APPROXIMATION
C
   27 G0=A*(1.25331414-.15666418*T(1)+.088111278*T(2)-.091390954*T(3)
     2+.13445962*T(4)-.22998503*T(5)+.37924097*T(6)-.52472773*T(7)
     3+.55753684*T(8)-.42626329*T(9)+.21845181*T(10)-.066809767*T(11)
     4+.009189383*T(12))*C
      IF(N)20,28,29
   28 BK=G0
      RETURN
C
C     COMPUTE K1 USING POLYNOMIAL APPROXIMATION
C
   29 G1=A*(1.2533141+.46999270*T(1)-.14685830*T(2)+.12804266*T(3)
     2-.17364316*T(4)+.28476181*T(5)-.45943421*T(6)+.62833807*T(7)
     3-.66322954*T(8)+.50502386*T(9)-.25813038*T(10)+.078800012*T(11)
     4-.010824177*T(12))*C
      IF(N-1)20,30,31
   30 BK=G1
      RETURN
C
C     FROM KO,K1 COMPUTE KN USING RECURRENCE RELATION
C
   31 DO 35 J=2,N
      GJ=2.*(DFLOAT(J)-1.)*G1/X+G0
      IF(GJ-1.0D70)33,33,32
   32 IER=4
      GO TO 34
   33 G0=G1
   35 G1=GJ
   34 BK=GJ
      RETURN
   36 B=X/2.
      A=.57721566+DLOG(B)
      C=B*B
      IF(N-1)37,43,37
C
C     COMPUTE KO USING SERIES EXPANSION
C
   37 G0=-A
      X2J=1.
      FACT=1.
      HJ=.0
      DO 40 J=1,6
      RJ=1./DFLOAT(J)
      X2J=X2J*C
      FACT=FACT*RJ*RJ
      HJ=HJ+RJ
   40 G0=G0+X2J*FACT*(HJ-A)
      IF(N)43,42,43
   42 BK=G0
```

```
                                                                        ISN
          RETURN                                           BESK1130     56
C                                                          BESK1140
C         COMPUTE K1 USING SERIES EXPANSION                BESK1150
C                                                          BESK1160
       43 X2J=B                                            BESK1170     57
          FACT=1.                                          BESK1180     58
          HJ=1.                                            BESK1190     59
          G1=1./X+X2J*(.5+A-HJ)                            BESK1200     60
          DO 50 J=2,8                                      BESK1210     61
          X2J=X2J*C                                        BESK1220     62
          RJ=1./DFLOAT(J)                                               63
          FACT=FACT*RJ*RJ                                  BESK1240     64
          HJ=HJ+RJ                                         BESK1250     65
       50 G1=G1+X2J*FACT*(.5+(A-HJ)*DFLOAT(J))                          66
          IF(N-1)31,52,31                                  BESK1270     67
       52 BK=G1                                            BESK1280     68
          RETURN                                           BESK1290     69
          END                                              BESK1300     70
```

```
                                                                        ISN
C
C***********************************
C
      SUBROUTINE BESINT(R,KAPPA,EPS)                                      1
C
C
C        BESINT INITIALIZES TABLES OF BESSEL FUNCTIONS FOR USE IN
C             DETERMINING THE POTENTIAL FUNCTION FOR THE HELMHOLTZ
C             PROBLEM
C
C          REF ABRAMOWITZ AND STEGEN, HANDBOOK OF MATHEMATICAL FUNCTIONS.
C
      IMPLICIT REAL*8(A-H,O-Z)                                            2
C
      REAL*4 SNGL                                                         3
      REAL*8 KAPPA,IO,IBES,KO,KBES,IOR,IBESR,KOR,KBESR,ION,IBESN,IONR     4
     1,IBESNR,KR
C
      COMMON / BESS / IO,IBES(91),KO,KBES(91),IOR,IBESR(90),KOR,          5
     1KBESR(90),ION,IBESN(91),IONR,IBESNR(90),NU
C
      DATA GAMMA / 0.57721566490153280D0 /                                6
      DATA PI / 3.14159265358979320D0 /                                   7
C
C
C***********************************************************************
C     NOTE:
C          THIS SUBROUTINE IS CALLED BY SKY TO GENERATE THE I BESSEL FUNCTIONS
C          AND DERIVATIVES WITH RESPECT TO ORDER AND ARGUMENT. THE K BESSEL
C          FUNCTIONS ARE NOT NEEDED. HENCE NUM WAS SET EQUAL TO 1. IF THE K
C          BESSEL FUNCTIONS ARE DESIRED, THEN SET NUM EQUAL TO THEDESIRED VALUE.
C***********************************************************************
C
      NUM=1                                                               8
      KR=KAPPA*R                                                          9
      NK=2*NUM                                                           10
C
C        FIND THE MAXIMUM ORDER NU
C
      D=DLOG(EPS)/KR                                                     11
      C=DLOG(DEXP(1.D0)/2.D0)                                            12
      B=D-2.D0-C                                                         13
      ALPHA=(-B+DSQRT(B**2-4.D0*D*(2.D0-C)))/(2.D0*(2.D0-C))             14
      CLN=DLOG(2.D0*PI*KR)                                               15
      IF(ALPHA.LT.1.5D0) ALPHA=1.5D0                                     16
      DO 10 I=1,20                                                       17
      ALN=DLOG(ALPHA)                                                    18
      F=-(CLN+ALN)/(2.D0*KR)+ALPHA*(C-ALN)-D                             19
      FP=-1.D0/(2.D0*KR*ALPHA)+C-1.D0-ALN                                20
      DEL=F/FP                                                           21
      ALPHA=ALPHA-DEL                                                    22
      IF(DABS(DEL).LE.00.01D0/KR) GO TO 12                               23
   10 CONTINUE                                                           24
      WRITE(6,1) KR,ALPHA,DEL                                            25
      STOP                                                               26
   12 GNU=ALPHA*KR                                                       27
      NU=IFIX(SNGL(GNU))+1                                               28
      IF(NU.LE.91) GO TO 15                                              29
```

```
      WRITE(6,2) KR,ALPHA,DEL,GNU,NU                          30
      STOP                                                    31
C
C        FIND I(N,KR) FOR (0<=N<=NU) BY USE OF RECURSION RELATIONS
C
15    IBES(NU)=0.D0                                           32
      IBES(NU-1)=EPS                                          33
      SUM=EPS                                                 34
      DO 20 N=3,NU                                            35
      NN=NU-N+1                                               36
      IBES(NN)=2.D0*DFLOAT(NN+1)/KR*IBES(NN+1)+IBES(NN+2)     37
20    SUM=SUM+IBES(NN)                                        38
      I0=2.D0*IBES(1)/KR+IBES(2)                              39
      SUM=I0+2.D0*SUM                                         40
C
C        NORMALIZE THE VALUES OF I(N,KR)
C
      FACT=DEXP(KR)/SUM                                       41
      I0=I0*FACT                                              42
      DO 22 N=1,NU                                            43
      IBES(N)=FACT*IBES(N)                                    44
22    CONTINUE                                                45
      NU=NU-5                                                 46
C
C        FIND DERIVATIVES W. R. T. ORDER (D/DN) OF I(N,KR) BY USING
C             EQUATION 9.6.42 FROM A & S
C
      B=0.5D0*KR                                              47
      C=B*B                                                   48
      F=0.D0                                                  49
      PSI=-GAMMA                                              50
      TERM=1.D0                                               51
      SUM=I0*DLOG(B)-PSI*TERM                                 52
      DO 26 K=1,50                                            53
      F=F+1.D0                                                54
      PSI=PSI+1.D0/F                                          55
      TERM=TERM*C/(F*F)                                       56
      SUM=SUM-PSI*TERM                                        57
      IF(DABS(SUM).GT.TERM*1.D16) GO TO 27                    58
26    CONTINUE                                                59
      K=0                                                     60
      WRITE(6,3) KR,SUM,TERM,N,NU                             61
      STOP                                                    62
27    ION=SUM                                                 63
      DO 35 N=1,NU                                            64
      PSI=-GAMMA                                              65
      G=0.D0                                                  66
      GK=B**N                                                 67
      SUM=IBES(N)*DLOG(B)                                     68
      DO 30 K=1,N                                             69
      G=G+1.D0                                                70
      GK=GK/G                                                 71
      PSI=PSI+1.D0/G                                          72
      IF(PSI*GK*1.D16.LT.DABS(SUM)) GO TO 35                  73
30    CONTINUE                                                74
      F=0.D0                                                  75
      TERM=GK                                                 76
      SUM=SUM-PSI*TERM                                        77
```

```
                                                                          ISN
          DO 32 K=1,NU                                                    78
          G=G+1.D0                                                        79
          F=F+1.D0                                                        80
          PSI=PSI+1.D0/G                                                  81
          TERM=TERM*C/(G*F)                                               82
          SUM=SUM-PSI*TERM                                                83
          IF(DABS(SUM).GT.TERM*1.D16) GO TO 35                            84
   32     CONTINUE                                                        85
          WRITE(6,3) KR,SUM,TERM,N,NU                                     86
          STOP                                                            87
   35     IBESM(N)=SUM                                                    88
C
C            FIND K(N,KR) FOR (0<=N<=2*NUM) BY USE OF RECURSION RELATIONS
C
          KO=-IOM                                                         89
          KBES(1)=-IO/KR+IDESM(1)                                         90
C
C            USE RECURSION TO FIND K(N,KR) FOR N>1
C
          KBES(2)=2.D0*KBES(1)/KR+KO                                      91
          DO 42 J=3,NK                                                    92
   42     KBES(J)=2.D0*DFLOAT(J-1)*KBES(J-1)/KR+KBES(J-2)                 93
C
C            FIND D/DR OF I(N,KR) & K(N,KR) BY RECURSION RELATION
C
          IOP=KAPPA*IBES(1)                                               94
          IBESR(1)=0.5D0*KAPPA*(IO+IBES(2))                               95
          NM1I=NI-1                                                       96
          DO 44 N=2,NM1I                                                  97
   44     IBESR(N)=0.5D0*KAPPA*(IBES(N-1)+IBES(N+1))                      98
          KOP=-KAPPA*KBES(1)                                              99
          KBESR(1)=-0.5D0*KAPPA*(KO+KBES(2))                              100
          NM1K=NK-1                                                       101
          DO 45 N=2,NM1K                                                  102
   45     KBESR(N)=-0.5D0*KAPPA*(KBES(N-1)+KBES(N+1))                     103
C
C            FIND D**2/DR*DN OF I(N,KR) BY RECURSION RELATIONS FOR D/DR
C                  AND THE PREVIOUSLY DEFINED VALUES OF D*I(N,KR)/DN
C
          IOMR=IDESM(1)*KAPPA                                             104
          IBESMR(1)=0.5D0*KAPPA*(IOM+IBESM(2))                            105
          DO 55 N=2,NM1I                                                  106
   55     IBESMR(N)=0.5D0*KAPPA*(IBESM(N-1)+IBESM(N+1))                   107
          RETURN                                                          108
    1     FORMAT('-***** ERROR NO CONVERGENCE IN BESINT ',3(D14.7,2X))    109
    2     FORMAT('-***** ERROR NO GREATER THAN 91 IN BESINT ',4(D14.7,2X),I 110
         1I5)
    3     FORMAT('-*****ERROR IN BESINT(IBESM)',3D14.7,2I5)               111
          END                                                             112
```

```
C
C**********************************
C
      DOUBLE PRECISION FUNCTION FNT1(U)
C
C          FNT1 IS THE FUNCTION CALLED BY THE DOL_ SUBROUTINES FROM *SSP. THE
C          DOL_ SUBROUTINES USE A LAGUERRE QUADRATURE TO INTEGRATE FNT1 FROM
C          ZERO TO INFINITY. FNT1(U) RETURNS THE K1 BESSEL FUNCTION MINUS ITS
C          1/ARG SINGULARITY. THE BK(I) VARIABLES ARE STORED TO BE USED BY FNT3.
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /TEST/ X,XP,Y,YP,K,BK(12),ICOUNT
      REAL*8 K,K1
      R1=DSQRT((X-XP)*(X-XP)+(U-Y-YP)*(U-Y-YP))
      ARG=K*R1
      ICOUNT=ICOUNT+1
      IF(R1 .LE. 1.D-10) GO TO 10
      CALL BK1SOD(ARG,K1,IER)
      BK(ICOUNT)=K1/R1
      FNT1=K1/R1
      RETURN
   10 BK(ICOUNT)=0.D0
      FNT1=BK(ICOUNT)
      RETURN
      END
```

```
C
C************************************
C
      DOUBLE PRECISION FUNCTION FNT2(U)
C
C     FNT2(U) IS SIMILAR TO FNT1, THOUGH IT RETURNS THE K1 BESSEL FUNCTION
C     WITH ITS 1/ARG SINGULARITY. AGAIN BK(I) IS STORED TO BE USED
C     WITH FNT3.
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /TEST/ X,XP,Y,YP,K,BK(12),ICOUNT
      REAL*8 K,K1
      R1=DSQRT((X-XP)*(X-XP)+(U-Y-YP)*(U-Y-YP))
      ARG=K*R1
      ICOUNT=ICOUNT+1
      IF (ARG .GT. 169.D0) GO TO 5
      CALL BESK(ARG,1,K1,IER)
      BK(ICOUNT)=K1/R1
      FNT2=K1/R1
      RETURN
    5 BK(ICOUNT)=0.D0
      FNT2=BK(ICOUNT)
      RETURN
      END
```

```
C
C**********************************
C
      DOUBLE PRECISION FUNCTION FNT3(U)
C
C        FNT3 IS SIMILAR TO FNT1 ANDFNT2. THE DIFFERENCE IS THAT FNT3 RETURNS
C        (U-Y-YP)*DK(I) INSTEAD OF JUST DK(I) AS FNT1 AND FNT2 DO.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /TEST/ X,XP,Y,YP,K,DK(12),ICOUNT
      REAL*8 K
      ICOUNT=ICOUNT+1
      FNT3=(U-Y-YP)*DK(ICOUNT)
      RETURN
      END
```

```
C
C*********************************
C
      SUBROUTINE PIK(X,VAL3)
C
C          BIK EVALUATES THE INTEGRAL OF KO(T) FOR T .GE. 0 AND .LE. X.
C          HERE KO IS THE K BESSEL FUNCTION OF ORDER 0. VAL3 IS THE RETURNED
C          VALUE OF THE INTEGRAL. THE MAXIMUM ERROR SHOULD BE LESS THAN 1.D-4.
C
      DIMENSION VAL1(51),COEF(7)
      DATA GAMMA /0.577217/
      DATA VAL1 /.71762,.705565,.694120,.683241,.672882,
     1           .663001,.653561,.644529,.635876,.627576,
     1           .619603,.611937,.604558,.597446,.590591,
     1           .583971,.577575,.571391,.565406,.559610,
     1           .553994,.548547,.543261,.538129,.533142,
     1           .528295,.523580,.518991,.514524,.510172,
     1           .505931,.501795,.497761,.493825,.489981,
     1           .486228,.482561,.478977,.475471,.472046,
     1           .468692,.465411,.462198,.459052,.455969,
     1           .452949,.449989,.447087,.444241,.441449,
     1           .438410/
      DATA COEF /1.25331E0,1.11902E-1,2.57664E-2,
     1           9.33994E-1,4.17454E-3,1.63271E-3,
     1           .033934E-3/
C
C          THE INTEGRAL IS EVALUATED AS FOLLOWS:
C               FOR X .LT. 2. - SERIES EXPANSION
C               FOR X .GE. 2 AND .LE. 7 - INTERPOLATED VALUES FROM A TABLE
C               FOR X .GE. 7 - SEVEN TERM CURVE FIT
C
      IF(X .GE. 7.0) GO TO 70
      IF(X .GE. 2.0) GO TO 40
      IF(X .LE. 1.E-7) GO TO 56
C
C          SERIES IS GIVEN ABRAMOWITZ AND STEGUN, 11.1.9.
C
      FACT=1.0
      X1=X/2.0
      SUM1=1.0
      SUM2=1.0
      SUM3=0.0
      X1SQ=X1*X1
      X1C=1.0
      COUNT=0.0
      FECO=0.0
      DO 10 I=1,5
      COUNT=COUNT+1.0
      FACT=FACT*COUNT
      X1C=X1C*X1SQ
      FTSQ=FACT*FACT
      DEN=2.0*COUNT+1.0
      FECO=FECO+1.0/COUNT
      SUM1=SUM1+X1C/(FTSQ*DEN)
      SUM2=SUM2+X1C/(FTSQ*DEN*DEN)
   10 SUM3=SUM3+X1C*FECO/(FTSQ*DEN)
      VAL3=-(GAMMA+ALOG(X1))*X*SUM1+X*SUM2+X*SUM3
      RETURN
```

```
C
C         FOR INTERPOLATED VALUES, VAL1 IS FROM A&B TABLE11.1. THIS IS THE
C         VALUE OF THE SAME INTEGRAND FOR A RANGE OF INTEGRATION FROM X TO
C         INFINITY, AND HENCE IS SUBTRACTED FROM THE INTEGRAL FROM 0 TO
C         INFINITY LEAVING THE DESIRED RESULT.
C
   40 ICOUNT=INT(10.0*(X-2.0))+1
      IF(ICOUNT .GE. 51) GO TO 45
      XP=2.0+FLOAT(ICOUNT-1)*0.1
      VAL2=VAL1(ICOUNT)-(VAL1(ICOUNT)-VAL1(ICOUNT+1))
     1                *(X-XP)/0.1
      GO TO 46
   45 VAL2=VAL1(51)
   46 VAL3=1.570796-EXP(-X)*VAL2
      RETURN
   56 VAL3=0.0
      RETURN
C
C         THIS CURVE FIT IS GIVEN IN A&B, 11.1.18. IT LIKE THE INTERPOLATED
C         VALUES, IS FOR A RANGE OF INTEGRATION FROM X TO INFINITY AND
C         HENCE MODIFIED ACCORDINGLY.
C
   70 X7=X/7.0
      CCN=1.0
      SUM1=CCEF(1)
      X7C=1.0
      DO 77 I=2,7
      X7C=X7C*X7
      CCN=-CCN
   77 SUM1=SUM1+CCN*CCEF(I)/X7C
      VAL3=1.570796-SUM1*EXP(-X)/SQRT(X)
      RETURN
      END
```

```
C
C**********************************
C
      SUBROUTINE S1Y (X,Y,K,S2,S2X,S2Y,EPS)
C
C          S1Y FINDS A SERIES THAT HELPS DEFINE THE SOURCE POTENTIAL FOR THE
C          HELMHOLTZ PROBLEM. S2 IS THE SUM OF NUM TERMS OF
C          ((-1)**N)*SINH(ALPHA*N)*D(IN(KR))*COS(N*ZETA))/DN
C          WHERE N RANGES FROM 1 TO NUM, AND IN IS THE BESSEL FUNCTION OF
C          ORDER N AND S2X, S2Y ARE THE DERIVATIVES OF S2 WITH RESPECT X AND Y.
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8      K,IO,IBES,KO,KBES,IOR,IBESR,KOR,KBESR,IOM,IBESM,IOMR
     1,IBESMR,KR
      COMMON / BESS / IO,IBES(91),KO,KBES(91),IOR,IBESR(90),KOR,
     1KBESR(90),IOM,IBESM(91),IOMR,IBESMR(90),NU
      ALPHA=DLOG(1.D0/K+DSQRT(1.D0/(K*K)-1.D0))
      R=DSQRT(X*X+Y*Y)
      ZETA=DATAN2(X,Y)
      RY=DCOS(ZETA)
      ZETAY=-DSIN(ZETA)/R
      RX=DSIN(ZETA)
      ZETAX=DCOS(ZETA)/R
C
C
C          BESINT COMPUTES THE I BESSEL FUNCTIONS AND THEIR DERIVATIVES WITH
C          RESPECT TO ARGUMENT AND ORDER.
      CALL BESINT(R,K,EPS)
      T1=0.D0
      T2=0.D0
      S2=0.D0
      CCN=1.D0
      NUM=NU-1
      CCU=0.D0
      DO 10 I=1,NUM
      CCU=CCU+1.D0
      CCN=-1.D0*CCN
      SIH=DSINH(CCU*ALPHA)
      CS=DCOS(CCU*ZETA)
      SI=DSIN(CCU*ZETA)
      T1=T1-CCN*(IBESMR(I)*CS-IBESR(I)*ZETA*SI)*SIH
      T2=T2+CCN*(IBESM(I)*SI*CCU+IBES(I)*(SI+CCU*ZETA*CS))*SIH
      S2=S2-CCN*(IBESM(I)*CS-IBES(I)*ZETA*SI)*SIH
   10 CONTINUE
      S2X=RX*T1+ZETAX*T2
      S2Y=RY*T1+ZETAY*T2
      RETURN
      END
```

```
                                                                          ISN
      C
      C**********************************
      C
            SUBROUTINE SANDC(X,BIGSI,BIGCI)                                 1
      C
      C          DOUBLE PRECISION SUBROUTINE TO EVALUATE THE SINE AND
      C          COSINE INTEGRALS
      C
            IMPLICIT REAL*8(A-H,O-Z)                                        2
            REAL*8 MULT                                      PEG  050       3
            DATA GAMMA,HALFPI/0.57721566,1.5707936/          PEG  060       4
            DATA A1,A2,A3,A4  1,B2,B3,B4/42.242855,302.757865,352.018498,21.82PEG  070   5
           11899,48.196927,482.485984,1114.978085,449.690326/  PEG  080
             DATA C1,C2,C3,C4,D1,D2,D3,D4/38.027264,265.187033,335.677320,38.1PEG  090   6
           102495,40.021433,322.624911,570.236280,157.105423/  PEG  100
            IF(X.LE.0.)GO TO 100                             PEG  110       7
            IF(X-1.)101,102,102                              PEG  120       8
      C     CALCULATION FOR 0. < X < 1.                      PEG  130
      C     CALCULATION FOR BIG CI(X)                        PEG  140
      101   SUM=0.                                           PEG  150       9
            FACT1=1.                                         PEG  160      10
            FACT2=1.                                         PEG  170      11
            MULT=1.                                          PEG  180      12
            DO 500 M=2,100,2                                 PEG  190      13
            FACT1=FACT1*X*X                                  PEG  200      14
            FACT2=FACT2*DFLOAT(M)*DFLOAT(M-1)                PEG  210      15
            MULT=MULT*(-1.)                                  PEG  220      16
            ADD=FACT1*MULT/(FACT2*DFLOAT(M))                 PEG  230      17
            IF(DABS(ADD).LT.0.00000001)GO TO 151             PEG  240      18
      500   SUM=SUM+ADD                                      PEG  250      19
      151   BIGCI=GAMMA+DLOG(X)+SUM                          PEG  260      20
      C     CALCULATION FOR BIG SI(X)                        PEG  270
            BIGSI=X                                          PEG  280      21
            FACT1=X                                          PEG  290      22
            FACT2=1.                                         PEG  300      23
            MULT=1.                                          PEG  310      24
            DO 501 M=3,100,2                                 PEG  320      25
            FACT1=FACT1*X*X                                  PEG  330      26
            FACT2=FACT2*DFLOAT(M)*DFLOAT(M-1)                PEG  340      27
            MULT=MULT*(-1.)                                  PEG  350      28
            ADD=FACT1*MULT/(FACT2*DFLOAT(M))                 PEG  360      29
            IF(DABS(ADD).LT.0.00000001)GO TO 152             PEG  370      30
      501   BIGSI=BIGSI+ADD                                  PEG  380      31
       152  RETURN                                                         32
      C     RATIONAL APPROXIMATION FOR X > OR = 1.           PEG  420
      102   DF=(X**8+C1*X**6+C2*X**4+C3*X**2+C4)/(X**8+D1*X**6+D2*X**4+D3*X**2PEG  430   33
           1+D4)/X                                           PEG  440
            DG=(X**8.0+A1*X**6.0+A2*X**4.0+A3*X*X+A4)/(X**8.0+B1*X**6.0+B2*X**PEG  450   34
           14.0+B3*X*X+B4)*(1./(X*X))                        PEG  460
            SINX=DSIN(X)                                                   35
            COSX=DCOS(X)                                                   36
            BIGSI=HALFPI-DF*COSX-DG*SINX                                   37
            BIGCI=DF*SINX-DG*COSX                                          38
            RETURN                                           PEG  470      39
      C     CASE OF X < OR = 0.                              PEG  480
      100   WRITE(6,800)X                                    PEG  490      40
      800   FORMAT(1H ,'X = ',G22.15,'; MUST BE GREATER THAN 0.'/'CALCULATION PEG  500   41
           1SUPPRESSED')                                     PEG  510
```

```
1677      DF=0.
1678      DG=0.
1679      RETURN
1680      END
```


ISN

42
43

```
C
C**********************************
C
      SUBROUTINE DEICOM(X1,Y1,EREAL,FIMG,IER)                          1
C
C
C     THIS SUBROUTINE CALCULATES THE COMPLEX EXPONENTIAL INTEGRAL
C     OF
C           EXP(-T)/T
C     FROM Z TO INFINITY.
C
C     WHERE Z=X1+I*Y1 AND Y1 .GE. 0 AND ABS(X1) .GE. 0 BUT
C     NOT Y1=0 AND X1=0.
C
C   REF. ABRAMOWITZ AND STEGUN, HANDBOOK OF MATHEMATICAL FUNCTIONS
C
C
C
      IMPLICIT REAL*8(A-H,O-Z)                                          2
      REAL*4 SNGL                                                       3
      DIMENSION A1(2),A2(2),A3(2),A4(2),A5(2)                           4
      DIMENSION B1(2),B2(2),B3(2),B4(2),B5(2),B6(2),B7(2),B8(2)         5
      DIMENSION W1(11)                                                  6
      DATA B1/3.69188589340D-1,0.170279632100D/,                        7
     1     B2/4.18786780F1D-1,0.903701776700/,
     2     B3/1.75794986630D-1,2.25109662980D0/,
     3     B4/3.33434922610D-2,4.26670017070D0/,
     4     B5/2.79453623520D-3,7.04590540210D0/,
     5     B6/9.07650877330D-5,10.758516010100/,
     6     B7/8.48574671620D-7,15.740678641200/,
     7     B8/1.04800117480D-9,22.8631317369D0/
      DATA A1/5.21755610520-1,0.26356031970D/,                          8
     1     A2/3.98666811080-1,1.41340305910D/,
     2     A3/7.59424496810-2,3.59642577100D/,
     3     A4/3.61175867990-3,7.08581000580D/,
     4     A5/2.33699723850-5,12.6408008442D0/
      DATA PI/3.14159265D/, W1/1.D0,4.D0,2.D0,4.D0,2.D0,4.D0,           9
     1 2.D0,4.D0,2.D0,4.D0,1.0D0/
      DATA GAMA/0.5772156649D0/                                         10
      IER=0                                                             11
      X=X1                                                              12
      Y=DABS(Y1)                                                        13
      IF(DABS(X) .LE. 1.D-6) GO TO 30                                   14
      IF(X .GE. 10.D0 .OR. X .GE. 0.D0 .AND. Y .GE.10.D0) GO TO 20      15
      IF( X .LE. -10.D0 .AND. Y .LE. 8.D0) GO TO 40                     16
      IF(DABS(X) .LT. 10.D0 .AND. Y .LT. 10.D0) GO TO 10                17
C*********  SEE NOTE BELOW  **********************************
C
C     IF(Y.LE. 1.D-6) GO TO 50
C
C*************************************************************
C
C           EIGHT POINT LAGUERRE INTEGRATION FOR X .LT. 0.0 AND
C           Y .GE. 10.0, OR 8.0 .LE. Y .LE. 10.0 AND X .LE. -10.0
C
      DB1=(B1(2)+X)*(B1(2)+X)+Y*Y                                       18
      DB2=(B2(2)+X)*(B2(2)+X)+Y*Y                                       19
      DB3=(B3(2)+X)*(B3(2)+X)+Y*Y                                       20
```

```
                                                          ISN
      DD4=(B4(2)+X)*(B4(2)+X)+Y*Y
      DD5=(B5(2)+X)*(B5(2)+X)+Y*Y
      DD6=(B6(2)+X)*(B6(2)+X)+Y*Y
      DD7=(B7(2)+X)*(B7(2)+X)+Y*Y
      DD8=(B8(2)+X)*(B8(2)+X)+Y*Y
      ER=B1(1)*(B1(2)+X)/DB1+
     1   B2(1)*(B2(2)+X)/DB2+
     1   B3(1)*(B3(2)+X)/DB3+
     1   B4(1)*(B4(2)+X)/DB4+
     1   B5(1)*(B5(2)+X)/DB5+
     1   B6(1)*(B6(2)+X)/DB6+
     1   B7(1)*(B7(2)+X)/DB7+
     1   B8(1)*(B8(2)+X)/DB8
      EI=-Y*(B1(1)/DB1+B2(1)/DB2+B3(1)/DB3+B4(1)/DB4+B5(1)/DB5+
     1 B6(1)/DB6+B7(1)/DB7+B8(1)/DB8)
      EX=DEXP(-X)
      CCSY=DCOS(Y)
      SINY=DSIN(Y)
      FREAL=EX*(ER*CCSY+EI*SINY)
      EIMG=EX*(EI*CCSY-ER*SINY)
      RETURN
C
C      SERIES EXPANSION FOR ABS(X)
C
C
   10 THETA=DATAN2(Y,X)
      R=DSQRT(X*X+Y*Y)
      CON=-GAMA-DLOG(R)
      RN=1.D0
      ER=X
      EI=Y
      SUM1=RN*ER
      SUM3=RN*EI
      DO 11 N=2,70
      RN=-RN*(N-1)/(N*N)
      ER2=ER*X-EI*Y
      EI2=ER*Y+EI*X
      SUM1=SUM1+RN*ER2
      SUM3=SUM3+RN*EI2
      IF(.  .(RN*ER2)    .LE.  DABS(CON+SUM1)*1.D-14)*1.D-7
     1 .AND. DABS(RN*EI2)    .LE. (DABS(-THETA+SUM3)*1.D-14)*
     2 1.D-7) GO TO 12
      ER=ER2
   11 EI=EI2
   12 FREAL=CON+SUM1
      EIMG=-THETA+SUM3
      RETURN
C
C        FIVE POINT LAGUERRE INTEGRATION FOR X .GT. 10. AND/OR
C        Y .GE. 10
C
   20 DA1=(A1(2)+X)*(A1(2)+X)+Y*Y
      DA2=(A2(2)+X)*(A2(2)+X)+Y*Y
      DA3=(A3(2)+X)*(A3(2)+X)+Y*Y
      DA4=(A4(2)+X)*(A4(2)+X)+Y*Y
      DA5=(A5(2)+X)*(A5(2)+X)+Y*Y
      ER=A1(1)*(A1(2)+X)/DA1+
     1   A2(1)*(A2(2)+X)/DA2+
```

```
     1   A3(1)*(A3(2)+X)/DA3+
     1   A4(1)*(A4(2)+X)/DA4+
     1   A5(1)*(A5(2)+X)/DA5
         EI=-Y*(A1(1)/DA1+A2(1)/DA2+A3(1)/DA3+A4(1)/DA3+A5(1)/DA5)     60
         EX=DEXP(-X)                                                    61
         COSY=DCOS(Y)                                                   62
         SINY=DSIN(Y)                                                   63
         ERFAL=EX*(ER*COSY+EI*SINY)                                     64
         EIMG=EX*(EI*COSY-ER*SINY)                                      65
         RETURN                                                         66
C
C         EXPONENTIAL INTEGRAL FOR IMAGINARY ARGUMENT
C
   30    CALL SANDC(Y,SI,CI)                                            67
         ERFAL=-CI                                                      68
         EIMG=SI-PI/2.D0                                                69
         RETURN                                                         70
C*********************************************************
C         NOTE:
C         IF THE VALUES OF DEI(X) (I.E. Y=0) ARE DESIRED
C         THEN THE FOLLOWING FORTRAN STATEMENTS SHOULD BE INCLUDED
C*********************************************************
C
C
C        THE EXPONENTIAL INTEGRAL FOR REAL ARGUMENTS
C
C
C  50    ERFAL=-DEI(-X)
         EIMG=0.D0
C        IF(X .LE. 0.D0) EIMG=-PI
C        RETURN
C
C
C              SIMPSONS INTEGRATION FOR X .LE. -10.0 AND Y .LE 8.0
C
   40    NC=IFIX(SNGL(Y))+1                                             71
         DELT=Y/(10.D0*DFLOAT(NC))                                      72
         CO=DCOS(DELT)                                                  73
         SI=DSIN(DELT)                                                  74
         FN= X*X+DELT*DELT                                              75
         SUM1=A1(2)*(X*SI+DELT*CO)/FN                                   76
         SUM2=-1.D0/X+A1(2)*(DELT*SI-X*CO)/FN                           77
         DO 47 I=1,NC                                                   78
         IF(I.EQ.1) GO TO 41                                            79
         I2=1                                                           80
         CN1=DCOS(TN-DELT)                                              81
         CN2=DCOS(TN-2.D0*DELT)                                         82
         SN1=DSIN(TN-DELT)                                              83
         SN2=DSIN(TN-2.D0*DELT)                                         84
         GO TO 45                                                       85
   41    CN2=1.D0                                                       86
         CN1=CO                                                         87
         SN2=0.D0                                                       88
         SN1=SI                                                         89
         I2=3                                                           90
   45    CONTINUE                                                       91
         DO 47 J=I2,11                                                  92
         TN=DFLOAT(J-1+(I-1)*10)*DELT                                   93
```

```
      RN=X*X+TN*TN
      CNN=2.D0*CN1*CC-CN2
      SNN=2.D0*SN1*CC-SN2
      SUM1=SUM1+W1(J)*(X*SNN+TN*CNN)/RN
      SUM2=SUM2+W1(J)*(TN*SNN-X*CNN)/RN
      IF(J .EQ. 11) GO TO 47
      CN2=CN1
      CN1=CNN
      SN2=SN1
      SN1=SNN
   47 CONTINUE
      EX=DEXP(-X)
      EIVAL=-DEI(-X)-EX*SUM1*DELT/3.D0
      EIMG=EX*SUM2*DELT/3.D0-PI
      RETURN
      END
```

```
C NAASA  2.1.001 DLUD     PIN-A 10-29-75      THE UNIV OF MICH COMP CTR
      SUBROUTINE DLUD (N,ADIM,A,TDIM,T,IV)                              1
C
C COMPUTES THE LU-DECOMPOSITION OF THE N X N MATRIX A USING
C GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING. THIS FACTOR-
C IZATION MAY BE EXPRESSED IN THE FORM
C           L(N-1)*P(N-1)*...*L(1)*P(1)*A = U.
C WHERE EACH L(J) IS THE IDENTITY MATRIX EXCEPT FOR THE SUB-
C DIAGONAL ELEMENTS IN COLUMN J, EACH P(J) IS A PERMUTATION
C MATRIX, AND U IS AN UPPER TRIANGULAR MATRIX.  THIS IS THE
C PREPARATORY STEP IN SOLVING A SYSTEM OF LINEAR EQUATIONS,
C INVERTING A MATRIX, OR CALCULATING A DETERMINANT. A DISCUSSION
C OF GAUSSIAN ELIMINATION AND THE LU-DECOMPOSITION AND THEIR
C RELATIONSHIP TO THE NUMERICAL SOLUTION OF SYSTEMS OF LINEAR
C EQUATIONS MAY BE FOUND IN EITHER WILKINSON (1965,CHAPTER 4)
C OR FORSYTHE AND MOLER (1967).
C
      INTEGER N,ADIM,TDIM,IV(1)                                         2
      DOUBLE PRECISION A(ADIM,N),T(TDIM,N)                              3
C
C   N  -> ORDER OF THE MATRIX A.
C ADIM -> ROW DIMENSION OF THE ARRAY A. BECAUSE A IS AN N X N
C          MATRIX, ADIM SHOULD NOT BE LESS THAN N.  IF ADIM IS LESS
C          THAN N, THE CONTENTS OF A ARE IGNORED, AND THE MATRIX
C          TO BE FACTORED IS ASSUMED TO BE STORED IN THE ARRAY T.
C          SINCE ADIM MUST BE A POSITIVE INTEGER, IT IS RECOMMENDED
C          THAT THE ACTUAL ARGUMENTS A AND T COINCIDE WHEN ADIM IS
C          LESS THAN N TO AVOID THE INCONSISTENCY WHICH ARISES WHEN
C          N EQUALS 1.
C   A  -> TWO-DIMENSIONAL ARRAY CONTAINING THE N X N MATRIX TO
C          BE FACTORED, I.E., THE COEFFICIENT MATRIX OF THE SYSTEM
C          OF LINEAR EQUATIONS OR THE MATRIX TO BE INVERTED. THE
C          CONTENTS OF A ARE NOT ALTERED.
C TDIM -> ROW DIMENSION OF THE ARRAY T.
C   T  <- TWO-DIMENSIONAL   RAY FOR RETURNING THE LU-DECOMPOSITION
C          OF A. THE SUBDIAGONAL ELEMENTS OF THE J-TH COLUMN OF THE
C          L(J) AND THE UPPER TRIANGULAR MATRIX U ARE RETURNED IN
C          THE CORRESPONDING ELEMENTS OF T.  IF ADIM IS LESS THAN N,
C          T MUST CONTAIN THE MATRIX TO BE FACTORED WHEN THIS SUB-
C          ROUTINE IS CALLED.
C  IV  <- VECTOR OF LENGTH N DEFINING THE PERMUTATION MATRICES
C          P(J): MULTIPLICATION ON THE LEFT BY P(J) INTERCHANGES
C          ROWS J AND IV(J).  IF IV(J) IS NOT EQUAL TO J, THEN
C          DET(A) = - DET(P(J)*A).  AND TO AID IN THE COMPUTATION
C          OF DET(A), IV(N) WILL CONTAIN +1 IF AN EVEN NUMBER OF
C          INTERCHANGES ARE PERFORMED AND -1 IF AN ODD NUMBER. THUS
C                DET(A) = IV(N)*T(1,1)* ... *T(N,N).
C          IV(N) WILL CONTAIN 0 IF A IS COMPUTATIONALLY SINGULAR.
C
      INTEGER I,J,K,KP1,L                                               4
      DOUBLE PRECISION TMP                                              5
      DOUBLE PRECISION PIV                                              6
C
C IF ADIM IS GREATER THAN OR EQUAL TO N, THE CONTENTS OF A ARE MOVED
C TO T; WHILE IF ADIM IS LESS THAN N, THIS INITIAL DATA MOVEMENT IS
C SKIPPED.  SINCE THE DATA MOVEMENT TIME IS PROPORTIONAL TO N**2 AND
C THE COMPUTATION TIME PROPORTIONAL TO N**3/3, SIGNIFICANT SAVINGS
C SHOULD NOT BE EXPECTED.
```

```
C
      IF ( ADIM .LT. N ) GO TO 8110
        DO 8100 J = 1, N
          DO 8100 I = 1, N
 8100          T(I,J) = A(I,J)
 8110 CONTINUE
C
C GAUSSIAN ELIMINATION CONSISTS OF N-1 STAGES. DURING THE K-TH
C STAGE, THE PERMUTATION MATRIX P(K), THE LOWER TRIANGULAR MATRIX
C L(K), AND THE K-TH ROW OF U ARE COMPUTED BASED ON THE CURRENT
C ELEMENTS OF THE K-TH RESIDUAL MATRIX, I.E., THE ELEMENTS T(I,J),
C I,J=K...N. ONLY THE ELEMENTS OF THE K-TH RESIDUAL MATRIX ARE
C REFERENCED DURING THE K-TH STAGE OF GAUSSIAN ELIMINATION.
C
      IV(N) = 1
      DO 8260 K = 1, N
          PIV = DABS(T(K,K))
          IF ( K .GE. N ) GO TO 8260
C
C SELECT THE PIVOT ROW FOR THE K-TH STAGE BY PARTIAL PIVOTING,
C I.E., THE MAXIMUM ELEMENT IN THE 1-ST COLUMN OF THE K-TH RESIDUAL
C MATRIX, AND SET IV(K) ACCORDINGLY. THE VARIABLE L HOLDS THE
C SUBSCRIPT OF THE PIVOT ROW, AND PIV THE ABSOLUTE VALUE OF THE
C PIVOT ELEMENT.
C
          L = K
          KP1 = K + 1
          DO 8200 I = KP1, N
              IF ( PIV .GE. DABS(T(I,K)) ) GO TO 8200
                PIV = DABS(T(I,K))
                L = I
 8200     CONTINUE
          IV(K) = L
C
C SAVE THE PIVOT ELEMENT IN TMP.  IF P(K) IS NONTRIVIAL, I.E., IV(K)
C IS NOT EQUAL TO K, THE PIVOT ELEMENT IS ALWAYS NONZERO; OTHERWISE,
C THE PIVOT ELEMENT MUST BE CHECKED.  IF THE PIVOT IS ZERO, I.E.,
C THE MATRIX IS COMPUTATIONALLY SINGULAR, THEN T(I,K) IS ZERO FOR
C I=K...N, AND THE COMPUTATION MAY PROCEED TO THE NEXT STAGE.
C
          TMP = T(L,K)
          IF ( K .NE. L ) GO TO 8210
            IF ( PIV .GT. 0.D0 ) GO TO 8220
              IV(N) = 0
              GO TO 8260
 8210     CONTINUE
            IV(N) = -IV(N)
            T(L,K) = T(K,K)
            T(K,K) = TMP
 8220     CONTINUE
C
C COMPUTE THE NONTRIVIAL ELEMENTS OF L(K).  BECAUSE OF THE PARTIAL
C PIVOTAL STRATEGY, THE ABSOLUTE VALUE OF L(I,K)=-T(I,K)/T(K,K) IS
C LESS THAN OR EQUAL TO 1. IF T(K) DENOTES THE K-TH RESIDUAL MATRIX,
C THEN THE SUBDIAGONAL ELEMENTS [illegible] 1-ST COLUMN OF THE MATRIX
C L(K) [illegible] T(K) ARE ZERO. THESE ELEMENTS ARE NOT ACTUALLY CAL[illegible]
C LATED, THEY ARE REPLACED BY THE ELEMENTS OF L(K).
C
```

```
                TMP = -TMP                                                        34
                DO 8230 I = KP1, N                                                35
      8230         T(I,K) = T(I,K) / TMP                                          36
      C
      C APPLY P(K) AND L(K) TO THE K-TH RESIDUAL MATRIX COLUMNWISE, I.E.,
      C FOR J=K+1...N, INTERCHANGE T(K,J) AND T(L,J), THE (K,J)-ELEMENT
      C OF U, AND THEN FOR I=K+1...N, REPLACE T(I,J) BY
      C             T(I,J) + L(K)(I,N) * T(K,J).
      C
                DO 8250 J = KP1, N                                                37
                   TMP = T(L,J)                                                   38
                   T(L,J) = T(K,J)                                                39
                   T(K,J) = TMP                                                   40
                   DO 8240 I = KP1, N                                             41
      8240            T(I,J) = T(I,J) + T(I,K) * TMP                              42
      8250      CONTINUE                                                          43
      8260 CONTINUE                                                               44
            IF (PIV .EQ. 0.D0) IV(N) = 0                                          45
            RETURN                                                                46
      C
      C FORSYTHE,G.E. AND MOLER,C.B. 1967. COMPUTER SOLUTION OF LINEAR
      C   ALGEBRAIC SYSTEMS. ENGLEWOOD CLIFFS,N.J.: PRENTICE-HALL.
      C WILKINSON,J.H. 1965. THE ALGEBRAIC EIGENVALUE PROBLEM. OXFORD:
      C   CLARENDON PRESS.
      C                        THE UNIVERSITY OF MICHIGAN COMPUTING CENTER
      C                        NUMERICAL ANALYSIS LIBRARY - JULY 1975
            END                                                                   47
```

```
C NAASA  2.1.003 DBS       FTN-A 10-29-75      THE UNIV OF MICH COMP CTR
      SUBROUTINE DBS (N,TDIM,T,IV,B)
C
C SOLVES THE SYSTEM OF LINEAR EQUATIONS AX=B, WHERE A DENOTES
C THE N X N COEFFICIENT MATRIX AND X AND B ARE N-VECTORS, BY
C BACK-SUBSTITUTION IN THE LU-DECOMPOSITION OF A. THIS
C DECOMPOSITION MUST BE PROVIDED IN THE ARRAY T AND VECTOR IV
C VIS-A-VIS THE SUBROUTINE DLUD. THE LU-DECOMPOSITION MAY BE
C EXPRESSED IN THE FORM
C              L(N-1)*P(N-1)*...*L(1)*P(1)*A = U,
C WHERE EACH L(J) IS THE IDENTITY MATRIX EXCEPT FOR THE SUB-
C DIAGONAL ELEMENTS IN COLUMN J, EACH P(J) IS A PERMUTATION
C MATRIX, AND U IS AN UPPER TRIANGULAR MATRIX. USING THIS
C NOTATION, THE BACK-SUBSTITUTION CONSISTS OF FORMING
C              Y = L(N-1)*P(N-1)*...*L(1)*P(1)*B
C AND SOLVING THE UPPER TRIANGULAR SYSTEM OF LINEAR EQUATIONS
C UX = Y, I.E., FOR I=N...1
C   X(I)=(Y(I)-U(I,N)*X(N)-...-U(I,I+1)*X(I+1))/U(I,I).
C THIS BACK-SUBSTITUTION YIELDS A VECTOR X WHICH IS THE EXACT
C SOLUTION OF A SYSTEM OF LINEAR EQUATIONS (A+F)X=B, WHERE
C //F// IS GENERALLY ON THE ORDER OF N*//A//*MACHEPS. THIS METHOD
C OF SOLVING SYSTEMS OF LINEAR EQUATIONS IS DESCRIBED IN BOTH
C WILKINSON (1965,CHAPTER 4) AND FORSYTHE AND MOLER (1967).
C
      INTEGER N,TDIM,IV(1)
      DOUBLE PRECISION T(TDIM,N),B(1)
C
C    N  -> ORDER OF THE SYSTEM OF LINEAR EQUATIONS.
C TDIM -> ROW DIMENSION OF THE ARRAY T.
C    T  -> TWO-DIMENSIONAL ARRAY CONTAINING THE LU-DECOMPOSITION
C          OF THE COEFFICIENT MATRIX VIS-A-VIS THE SUBROUTINE DLUD.
C   IV  -> VECTOR CONTAINING THE INTERCHANGE INFORMATION GENERATED
C          BY THE SUBROUTINE DLUD DURING THE COMPUTATION OF THE
C          LU-DECOMPOSITION OF THE COEFFICIENT MATRIX.
C    B  -- VECTOR CONTAINING THE RIGHT-HAND SIDE OF THE SYSTEM OF
C          LINEAR EQUATIONS. THE CONTENTS OF B ARE REPLACED BY THE
C          ELEMENTS OF THE SOLUTION.
C
      INTEGER I,K,NP1,L
      DOUBLE PRECISION TMP
C
C REPLACE THE CONTENTS OF THE VECTOR B BY THE VECTOR
C          ( L(N-1) (P(N-1) ... ( L(1) (P(1) B)...).
C THIS COMPUTATION IS PERFORMED IN N-1 STAGES, WHERE DURING THE
C K-TH STAGE, THE CONTENTS OF B ARE REPLACED BY L(K)( P(K) B ).
C THE PERMUTATION MATRICES P(K) ARE ... BY THE VECTOR IV, I.E.,
C MULTIPLICATION BY P(K) SIMPLY INTERCHANGES THE K-TH AND IV(K)-TH
C ELEMENTS OF THE VECTOR.  THE LOWER TRIANGULAR MATRIX L(K) IS THE
C IDENTITY MATRIX EXCEPT FOR THE SUBDIAGONAL ELEMENTS OF THE K-TH
C COLUMN, WHICH ARE STORED IN THE CORRESPONDING ELEMENTS OF THE
C ARRAY T.
C
      DO 8110 K = 1, N
         IF ( K .GE. N ) GO TO 8110
         L = IV(K)
         TMP = B(L)
         B(L) = B(K)
         B(K) = TMP
```

```
            KP1 = K + 1
            DO 8100 I = KP1, N
8100           B(I) = B(I) + T(I,K) * TMP
8110 CONTINUE
C
C REPLACE THE CONTENTS OF THE VECTOR B BY THE SOLUTION TO THE SYSTEM
C OF LINEAR EQUATIONS WITH UPPER TRIANGULAR COEFFICIENT MATRIX U
C AND RIGHT-HAND SIDE VECTOR B.  THE USUAL FORMULAS FOR THE BACK-
C SUBSTITUTION, WHICH ARE BASED ON THE SUCCESSIVE ROWS OF THE MATRIX
C AND ARE SUITABLE WHEN INNER-PRODUCTS ARE ACCUMULATED, ARE NOT
C EMPLOYED.  THE COMPUTATION HAS INSTEAD BEEN ARRANGED TO REFERENCE
C THE SUCCESSIVE COLUMNS OF U.  THUS AFTER B(I) HAS BEEN COMPUTED,
C IT IS REMOVED FROM THE SYSTEM BY SUBTRACTING B(I) TIMES THE I-TH
C COLUMN OF U FROM THE RESIDUAL VECTOR B(1)...B(I-1).
C
        K = N
8200        B(K) = B(K) / T(K,K)
            IF ( K .LE. 1 ) RETURN
            TMP = -B(K)
            KP1 = K
            K = K - 1
            DO 8210 I = 1, K
8210           B(I) = B(I) + T(I,KP1) * TMP
            GO TO 8200
C
C FORSYTHE,G.E. AND MOLER,C.B. 1967. COMPUTER SOLUTION OF LINEAR
C   ALGEBRAIC SYSTEMS. ENGLEWOOD CLIFFS,N.J.: PRENTICE-HALL.
C WILKINSON,J.H. 1965. THE ALGEBRAIC EIGENVALUE PROBLEM. OXFORD:
C   CLARENDON PRESS.
C
C                         THE UNIVERSITY OF MICHIGAN COMPUTING CENTER
        END               NUMERICAL ANALYSIS LIBRARY - JULY 1975
```

-91-

T A B L E   O F   C O N T E N T S
(ALPHABETICAL)

AD-A079 316 MICHIGAN UNIV ANN ARBOR DEPT OF NAVAL ARCHITECTURE --ETC F/G 20/4
HEADSEAS WAVE DIFFRACTION COMPUTER PROGRAM. USER MANUAL.(U)
AUG 79 R F BECK N00014-78-C-0109

UNCLASSIFIED NL

2 OF 2
AD
A079316

END
DATE
FILMED
2-80
DDC

T A B L E   O F   C O N T E N T S
(SEQUENTIAL)

APPENDIX II

Input Listing for Ore Carrier S.J. Cort

```
      1          20    30     2     1     2     1    11
      2           1.99       32.174     15.0         1.0
      3          0.0          0.128
      4          0.750
      5           4    11
END CF FILE
```

```
  1     0.0
-0.0          -0.0
 10     0.1875
-0.2491       -0.0          0
-0.2438       -0.0263       0
-0.2395       -0.0713       0
-0.2332       -0.1163       0
-0.2226       -0.1613       0
-0.2094       -0.2063       0
-0.1961       -0.2513       0
-0.1749       -0.2963       0
-0.1378       -0.3413       0
-0.0          -0.3863       0
  9     0.375
-0.4275       -0.0          0
-0.4150       -0.0529       0
-0.3933       -0.1363       0
-0.3642       -0.2196       0
-0.3175       -0.3029       0
-0.2500       -0.3583       0
-0.1700       -0.3863       0
-0.0833       -0.3863       0
-0.0          -0.3863       0
 10     0.750
-0.6667       -0.0          0
-0.6550       -0.0529       0
-0.6317       -0.1363       0
-0.5975       -0.2196       0
-0.5458       -0.3029       0
-0.4125       -0.3863       0
-0.3333       -0.3863       0
-0.2083       -0.3863       0
-0.1042       -0.3863       0
-0.0          -0.3863       0
 11     1.125
-0.7717       -0.0          0
-0.7658       -0.0529       0
-0.7525       -0.1363       0
-0.7317       -0.2196       0
-0.6933       -0.3029       0
-0.5867       -0.3863       0
-0.5000       -0.3863       0
-0.3750       -0.3863       0
-0.2500       -0.3863       0
-0.1250       -0.3863       0
-0.0          -0.3863       0
 12     1.500
-0.7845       -0.0          0
-0.7845       -0.0529       0
-0.7845       -0.1363       0
-0.7792       -0.2196       0
-0.7617       -0.3029       0
-0.6717       -0.3863       0
-0.5833       -0.3863       0
-0.5000       -0.3863       0
-0.3750       -0.3863       0
-0.2500       -0.3863       0
-0.1250       -0.3863       0
-0.0          -0.3863       0
```

60	10	2.250	
61	-0.7845	-0.0	0
62	-0.7845	-0.0833	0
63	-0.7845	-0.1667	0
64	-0.7845	-0.2600	0
65	-0.7845	-0.3671	0
66	-0.7800	-0.3800	0
67	-0.7653	-0.3863	0
68	-0.5000	-0.3863	0
69	-0.2500	-0.3863	0
70	-0.0	-0.3863	0
71	999	3.000	
72	999	4.500	
73	999	6.000	
74	999	7.500	
75	999	9.000	
76	999	10.500	
77	999	12.000	
78	999	12.750	
79	12	13.500	
80	-0.7845	-0.0	0
81	-0.7845	-0.0833	0
82	-0.7845	-0.1667	0
83	-0.7845	-0.2971	0
84	-0.7167	-0.3721	0
85	-0.6250	-0.3733	0
86	-0.5000	-0.3750	0
87	-0.3750	-0.3767	0
88	-0.2500	-0.3796	0
89	-0.1667	-0.3808	0
90	-0.0775	-0.3821	0
91	-0.0	-0.3863	0
92	12	13.875	
93	-0.7845	-0.0	0
94	-0.7845	-0.0583	0
95	-0.7845	-0.1167	0
96	-0.7845	-0.1746	0
97	-0.7742	-0.2196	0
98	-0.7167	-0.2471	0
99	-0.6067	-0.2558	0
100	-0.5000	-0.2646	0
101	-0.3750	-0.2750	0
102	-0.2500	-0.2862	0
103	-0.0983	-0.2987	0
104	-0.0	-0.3075	0
105	21	14.250	
106	-0.7683	-0.0	0
106.2	-0.7675	-0.0133	
107	-0.7667	-0.0267	0
107.2	-0.7662	-0.0398	
108	-0.7658	-0.0529	0
108.2	-0.7545	-0.0760	
109	-0.7433	-0.0992	0
109.2	-0.7254	-0.1085	
110	-0.7075	-0.1179	0
110.2	-0.6566	-0.1227	
111	-0.6058	-0.1275	0
111.2	-0.5563	-0.1319	
112	-0.5067	-0.1363	0
112.2	-0.4425	-0.1427	

113	-0.3783	-0.1492	0
113.2	-0.3142	-0.1552	
114	-0.2500	-0.1612	0
114.2	-0.1854	-0.1671	
115	-0.1209	-0.1729	0
115.2	-0.0604	-0.1781	
116	-0.0	-0.1833	0
117	17 14.625		
118	-0.6708	-0.0	0
118.2	-0.6633	-0.0027	
119	-0.6558	-0.0054	0
119.2	-0.6471	-0.0065	
120	-0.6383	-0.0075	0
120.2	-0.6033	-0.0113	
121	-0.5683	-0.0150	0
121.2	-0.5342	-0.0194	
122	-0.5000	-0.0237	0
122.2	-0.4396	-0.0310	
123	-0.3792	-0.0383	0
123.2	-0.3167	-0.0456	
124	-0.2542	-0.0529	0
124.2	-0.1880	-0.0611	
125	-0.1217	-0.0692	0
125.2	-0.0609	-0.0771	
126	-0.0	-0.0850	0
127	8 15.000		
128	-0.2817	-0.0	0
129	-0.2500	-0.0050	0
130	-0.2100	-0.0106	0
131	-0.1642	-0.0192	0
132	-0.1175	-0.0267	0
133	-0.0725	-0.0350	0
134	-0.0392	-0.0406	0
135	-0.0	-0.0475	0
136	1 15.0		
137	-0.0	-0.0	

END OF FILE

APPENDIX III

Output Listing for Ore Carrier S.J. Cort

```
  INPUT [illegible]
NSTA=   2[illegible]   [illegible]   30   [illegible]
[illegible]=   [illegible]   2   [illegible]   1   [illegible]   11
  [illegible]=   1.[illegible]   [illegible]   [illegible]   X[illegible]   [illegible].0000
 [illegible]=   [illegible]
[illegible]
  0.0         0.1[illegible]
WAVE LENGTHS ([illegible]/LBP)
  0.75[illegible]0
STATIONS AT [illegible] PRESSURE [illegible] DESIRED
  4   11
 [illegible]=   0.[illegible]E+02  BEA[illegible]=   0.1[illegible]E+01
```

WLVL= 0.7500 WAVEN= 0.5545 OMEGA= 4.2300

TWO DIMENSIONAL SOURCE STRENGTH DISTRIBUTION

XI	BO
0.0	0.0
0.0411	-0.14116E-01
0.1875	-0.57743E-01
0.3750	-0.98821E-01
0.6484	-0.10266E+00
0.7500	-0.15472E+00
1.0049	-0.17472E+00
1.1250	-0.17644E+00
1.5000	-0.16884E+00
1.9264	-0.16269E+00
2.2500	-0.15976E+00
2.4915	-0.15976E+00
3.0000	-0.15976E+00
3.0916	-0.15976E+00
3.7500	-0.15976E+00
4.5000	-0.15976E+00
5.1824	-0.15976E+00
6.0000	-0.15976E+00
6.7150	-0.15976E+00
7.5000	-0.15976E+00
8.2500	-0.15976E+00
9.0000	-0.15976E+00
9.9176	-0.15976E+00
10.5000	-0.15976E+00
11.2500	-0.15976E+00
11.9033	-0.15976E+00
12.0000	-0.15976E+00
12.5185	-0.15976E+00
12.7500	-0.15976E+00
13.0736	-0.16400E+00
13.5000	-0.17746E+00
13.8750	-0.18530E+00
13.9952	-0.19555E+00
14.2500	-0.22016E+00
14.3516	-0.22363E+00
14.6250	-0.20467E+00
14.9361	-0.14533E+00
15.0000	-0.69463E-01

FROU[illegible] = 0.0 [illegible]ITY = 0.0
ONEGA[illegible] = [illegible] TAU = 0.[illegible] TAU[illegible] = 0.0

[illegible] SOURCE [illegible] DISTRIBUTION

XI	[illegible](SIGMA)	[illegible](SIGMA0)	[illegible](SIGMA[illegible])	[illegible]
0.0	0.0	0.0	0.0	0.0
0.0411	0.33563E+00	0.44220E-01	0.43207E-01	0.10331E-[illegible]
0.1875	0.13397E+01	0.17638E+00	0.17531E+00	0.19447E-01
0.3750	0.21835E+01	0.29834E+00	0.23259E+00	0.57355E-01
0.6484	0.29322E+01	0.38642E+00	0.36905E+00	0.1145[illegible]E+00
0.7500	0.30240E+01	0.4[illegible]E+00	0.38562E+00	0.132[illegible]E+00
1.0[illegible]	0.2[illegible]70E+01	0.4[illegible]E+00	0.4[illegible]E+00	0.1[illegible]E+00
1.1250	0.28655E+01	0.[illegible]773E+00	0.3[illegible]E+00	0.1[illegible]E+00
1.5000	0.28322E+01	0.39370E+00	0.36169E+00	0.15[illegible]E+00
1.9264	0.27617E+01	0.36386E+00	0.33316E+00	0.14629E+00
2.2500	0.26514E+01	0.34334E+00	0.31883E+00	0.14265E+00
2.4815	0.26238E+01	0.34371E+00	0.3126[illegible]E+00	0.14[illegible]E+00
3.[illegible]	0.2[illegible]E+01	0.32213E+00	0.[illegible]E+00	0.14[illegible]E+00
[illegible]	0.[illegible]55E+01	0.33324E+00	0.2[illegible]E+00	0.14[illegible]E+00
[illegible]	0.[illegible]2E+01	0.31732E+00	0.2[illegible]E+00	0.14170E+00
4.5[illegible]	0.23202E+01	0.30559E+00	0.27142E+00	0.14063E+00
5.13[illegible]	0.22472E+01	0.29619E+00	0.26114E+00	0.1[illegible]1[illegible]E+00
6.0[illegible]	0.21721E+01	0.28[illegible]E+00	0.25[illegible]E+00	0.13[illegible]E+00
6.7[illegible]	0.21100E+01	0.273[illegible]E+00	0.2[illegible]E+00	0.1[illegible]78E+00
7.[illegible]	0.20515E+01	0.2791[illegible]E+00	0.2[illegible]91E+00	0.135[illegible]E+00
[illegible]	0.19365E+01	0.26[illegible]E+00	0.22[illegible]E+00	0.133[illegible]E+00
[illegible]	0.19513E+01	0.25[illegible]E+00	0.22[illegible]E+00	0.13[illegible]E+00
[illegible]176	0.19033E+01	0.25[illegible]E+00	0.21[illegible]E+00	0.131[illegible]E+00
10.5[illegible]	0.18659E+01	0.24527E+00	0.2[illegible]E+00	0.12[illegible]E+00
11.2[illegible]	0.18[illegible]1E+01	0.24[illegible]E+00	0.20[illegible]E+00	0.12[illegible]E+00
11.9[illegible]	0.17979E+01	0.23686E+00	0.1[illegible]E+00	0.125[illegible]E+00
12.0000	0.17936E+01	0.23[illegible]1E+00	0.1[illegible]E+00	0.12735E+00
12.5185	0.17795E+01	0.23326E+00	0.19[illegible]E+00	0.12648E+00
12.7500	0.17602E+01	0.23121E+00	0.19463E+00	0.12611E+00
13.0736	0.17939E+01	0.23625E+00	0.19[illegible]E+00	0.13[illegible]E+00
13.5000	0.18904E+01	0.24775E+00	0.20313E+00	0.14175E+00
13.8750	0.18993E+01	0.25820E+00	0.20580E+00	0.149[illegible]E+00
13.9952	0.19947E+01	0.26150E+00	0.20974E+00	0.15617E+00
14.2500	0.21532E+01	0.28435E+00	0.22103E+00	0.178[illegible]E+00
14.3516	0.21649E+01	0.29522E+00	0.22059E+00	0.18[illegible]E+00
14.6250	0.19754E+01	0.26227E+00	0.20542E+00	0.15[illegible]E+00
14.8361	0.14732E+01	0.19410E+00	0.16333E+00	0.1[illegible]E+00
15.0000	0.74729E+00	0.98459E-01	0.84193E-01	0.81635E-01

PRESSURE DISTRIBUTION FOR STATION 4 X= 0.7500

			2-D POTENTIAL		3-D PRESSURE	
Y	Z	THETA	[illegible]	[illegible]	[illegible](PRES)	[illegible](PRES)
-0.6667	-0.0	90.0000	1.0001	0.1575	0.1707	-95.[illegible]
-0.6550	-0.0529	95.3825	0.97[illegible]	0.1440	0.9350	-95.[illegible]
-0.6317	-0.1363	77.3240	0.[illegible]	0.1182	0.8763	-95.[illegible]
-0.5975	-0.2126	69.[illegible]	0.[illegible]	0.0[illegible]4	0.8135	-95.[illegible]
-0.5454	-0.2629	60.0712	0.[illegible]84	0.0[illegible]3	0.7426	-95.[illegible]
-0.4125	-0.2863	46.8[illegible]	0.[illegible]	-0.0[illegible]51	0.6464	-95.[illegible]
-0.3333	-0.2863	40.7876	0.[illegible]	-0.0[illegible]	0.[illegible]15[illegible]	-95.[illegible]
-0.2033	-0.[illegible]3	29.32[illegible]	0.[illegible]	-0.1[illegible]	0.[illegible]1[illegible]	-95.[illegible]
-0.1033	-0.[illegible]3	15.0[illegible]	0.[illegible]	-0.1132	0.5310	-[illegible]
-0.0	-0.[illegible]63	0.0	0.[illegible]9	-0.1171	0.5777	-[illegible]

PRESSURE DISTRIBUTION FOR STATION 11 X= 7.5000

Y	Z	THETA	2-D POTENTIAL REAL	PHRE	3-D PRESSURE MAG(PRES)	ARG(PRES)
-0.7845	-0.0	90.0000	1.0000	0.2681	0.6825	60.0649
-0.7845	-0.0833	83.[illegible]	0.9545	0.2526	0.6438	60.0649
-0.7845	-0.1667	78.0045	0.9111	[illegible]	[illegible]	[illegible]
-0.7845	-0.2600	71.6637	0.8683	0.1963	0.5712	60.0649
-0.7845	-0.3671	[illegible]	0.8156	0.1131	0.4994	60.0649
-0.7820	-0.3899	64.0256	0.8098	0.0927	0.4352	60.0649
-0.7652	-0.3963	62.0167	0.7959	0.0989	0.4870	60.0649
-0.5000	-0.3963	52.3103	0.8059	-0.0315	0.3999	60.0649
-0.2500	-0.3963	32.0225	0.8959	-0.1457	0.3554	60.0649
-0.0	-0.3963	0.0	0.8959	-0.1634	0.3453	60.0649

NONDIMENSIONAL WAVE AMPLITUDE (INCIDENT+DIFFRACTED) ALONG SHIP

XAXIS	MAGNITUDE	PHASE
0.0	1.0000	-90.0000
0.1875	0.9647	-89.6669
0.3750	0.9648	-90.5266
0.7500	0.9707	-95.0729
1.1250	0.9599	-103.2835
1.5000	0.9368	-114.7083
2.2500	0.8826	-137.0836
3.0000	0.8391	-160.5574
4.5000	0.7723	153.3994
6.0000	0.7224	106.0772
7.5000	0.6825	60.0649
9.0000	0.6495	13.[illegible]
10.5000	0.6214	-34.1151
12.0000	0.5973	-81.[illegible]
12.7500	0.5859	-105.[illegible]
13.[illegible]	0.5707	-127.[illegible]
13.[illegible]	0.5465	-13[illegible]
14.2500	0.5013	-147.[illegible]
14.[illegible]	0.4373	-16[illegible]
15.0000	0.4294	17[illegible]

EXCITING FORCE DISTRIBUTION

X	REAL(F3)	IMAG(F3)	MAG(F3)	ARG(F3)
0.0	0.0	0.0	0.0	0.0
0.0411	0.12277E-01	-0.49930E+00	0.49845E+00	-88.5986
0.1875	0.16606E+00	-0.14969E+01	0.15061E+01	-83.6699
0.3750	0.28736E+00	-0.14159E+01	0.14447E+01	-78.5265
0.5625	0.39933E+00	-0.12851E+01	0.13456E+01	-72.7546
0.7500	0.42513E+00	-0.12339E+01	0.13106E+01	-71.0723
1.0000	0.46022E+00	-0.11434E+01	0.12322E+01	-68.0829
1.1250	0.46193E+00	-0.11080E+01	0.12013E+01	-67.3835
1.5000	0.43711E+00	-0.10150E+01	0.11053E+01	-66.7040
1.9200	0.42350E+00	-0.96493E+00	0.10534E+01	-66.2937
2.2500	0.41902E+00	-0.93658E+00	0.10261E+01	-65.9886
2.4815	0.41914E+00	-0.91850E+00	0.10096E+01	-65.4712
3.0000	0.41850E+00	-0.89120E+00	0.97559E+00	-64.5077
3.3316	0.41328E+00	-0.37523E+00	0.97009E+00	-64.4565
3.7500	0.41623E+00	-0.83560E+00	0.93355E+00	-63.5212
4.5000	0.41392E+00	-0.79720E+00	0.89792E+00	-62.6097
5.1000	0.40922E+00	-0.76730E+00	0.86659E+00	-61.9871
6.0000	0.40562E+00	-0.73541E+00	0.83985E+00	-61.1229
6.7100	0.40172E+00	-0.71120E+00	0.81659E+00	-60.5251
7.5000	0.39756E+00	-0.69690E+00	0.79356E+00	-59.9350
8.2500	0.39333E+00	-0.66512E+00	0.77271E+00	-59.4319
9.0000	0.38953E+00	-0.64637E+00	0.75517E+00	-58.9530

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
11.[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
11.[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
12.5[illegible]	.27153E+00	-0.57572E+00	[illegible]	[illegible]
12.75[illegible]	0.27043E+00	-0.5717[illegible]E+00	0.68121E+00	-57.0531
13.07[illegible]	0.27663E+00	-0.56305E+00	0.62156E+00	-56.[illegible]
13.5[illegible]	0.29357E+00	-0.56[illegible]E+00	0.63788E+00	-55.0[illegible]
13.87[illegible]	0.30999E+00	-0.5655[illegible]E+00	0.6[illegible]52E+00	-54.05[illegible]
13.99[illegible]	0.42410E+00	-0.56[illegible]E+00	0.71[illegible]27E+00	-53.3[illegible]
14.25[illegible]	0.45839E+00	-0.566[illegible]E+00	0.7[illegible]47E+00	-51.014[illegible]
14.[illegible]	0.46032E+00	-0.5615[illegible]E+00	0.7[illegible]12E+00	-50.6[illegible]
14.[illegible]	0.48628E+00	-0.5736[illegible]E+00	0.72[illegible]E+00	-52.[illegible]
14.[illegible]	.29856E+00	-0.62081E+00	0.73774E+00	-57.2[illegible]
15.0[illegible]	0.31548E+00	-0.67509E+00	0.74517E+00	-64.0522

TOTAL [illegible] FORCES AND B.M. AT Y= 7.5000

	REAL	IMAG	MAG	PHASE
HEAVE=	[illegible]	-0.66312E-[illegible]	0.[illegible]E-01	[illegible]
PITCH=	-0.[illegible]46E-01	0.10672E-02	0.[illegible]066E-01	177.8208
B.M.=	[illegible]	-0.25074E-01	0.[illegible]E-01	-1[illegible]1.5[illegible]

FROUDE NUM= 0.1280 VELOCITY= 2.8120
OMEGAR= 5.8095 TAU= 0.5977 TAU*= 0.3705

THREE DIMENSIONAL SOURCE STRENGTH DISTRIBUTION

XI	MAG(SIGMA)	MAG(SIGMAND)	RE(SIGMAND)	IMAG(SIGMAND)
0.0	0.0	0.0	0.0	0.0
0.0411	0.33076E+00	0.43579E-01	0.43579E-01	0.80036E-04
0.1875	0.12733E+01	0.16776E+00	0.16774E+00	0.26571E-02
0.3750	0.20421E+01	0.26905E+00	0.26885E+00	0.10259E-01
0.64[illegible]	0.27271E+01	0.35931E+00	0.35851E+00	0.25433E-01
0.7500	0.28777E+01	0.38047E+00	0.37917E+00	0.31410E-01
1.00[illegible]	0.31032E+01	0.40965E+00	0.40723E+00	0.44557E-01
1.1250	0.30973E+01	0.40815E+00	0.40526E+00	0.48511E-01
1.5000	0.29134E+01	0.38454E+00	0.38072E+00	0.54758E-01
1.92[illegible]	0.27437E+01	0.36215E+00	0.35741E+00	0.58436E-01
2.2500	0.26664E+01	0.35131E+00	0.34600E+00	0.60842E-01
2.4815	0.26333E+01	0.34773E+00	0.34193E+00	0.63094E-01
3.0000	0.25830E+01	0.34031E+00	0.33359E+00	0.67329E-01
3.0916	0.25737E+01	0.33933E+00	0.33219E+00	0.68015E-01
3.7500	0.25115E+01	0.33090E+00	0.32298E+00	0.72424E-01
4.5000	0.24422E+01	0.32[illegible]E+00	0.31343E+00	0.76511E-01
5.1824	0.23984E+01	0.31620E+00	0.30581E+00	0.79593E-01
6.0000	0.23435E+01	0.30876E+00	0.29749E+00	0.82650E-01
6.7160	0.22936E+01	0.30223E+00	0.29084E+00	0.84323E-01
7.5000	0.22554E+01	0.29715E+00	0.28412E+00	0.87042E-01
8.2830	0.22145E+01	0.29176E+00	0.27791E+00	0.88846E-01
9.0000	0.21795E+01	0.28715E+00	0.27259E+00	0.90280E-01
9.9176	0.21422E+01	0.28224E+00	0.26693E+00	0.91636E-01
10.5000	0.21128E+01	0.27837E+00	0.26247E+00	0.92714E-01
11.2500	0.20922E+01	0.27433E+00	0.25784E+00	0.93733E-01
11.9024	0.20566E+01	0.27096E+00	0.25396E+00	0.94475E-01
12.0000	0.20531E+01	0.27050E+00	0.25343E+00	0.94574E-01
12.5185	0.20339E+01	0.26798E+00	0.25053E+00	0.95090E-01
12.7500	0.20254E+01	0.26686E+00	0.24925E+00	0.95336E-01
13.0736	0.20659E+01	0.27219E+00	0.25379E+00	0.98400E-01
13.5000	0.21697E+01	0.28587E+00	0.26559E+00	0.10572E+00
13.8750	0.22358E+01	0.29457E+00	0.27270E+00	0.11133E+00
13.9950	0.23005E+01	0.30311E+00	0.28019E+00	0.11562E+00
14.2500	0.25071E+01	0.33032E+00	0.30391E+00	0.12940E+00
14.3516	0.25243E+01	0.33259E+00	0.30543E+00	0.13162E+00
14.6250	0.23302E+01	0.30702E+00	0.28137E+00	0.12290E+00
14.8361	0.17630E+01	0.23224E+00	0.21429E+00	0.91315E-01
15.0000	0.92653E+00	0.12207E+00	0.11326E+00	0.45545E-01

PRESSURE DISTRIBUTION FOR STATION 4 Y= 0.7500

			2-D POTENTIAL		2-D PRESSURE	
Y	Z	THETA	[illegible]	[illegible]	MAG(PRES)	ARG(PRES)
-0.6667	-0.0	90.0000	1.0000	0.1575	0.9163	-100.2660
-0.6550	-0.0529	85.2225	0.9700	0.1440	0.8727	-100.2660
-0.6317	-0.1363	77.9240	0.9267	0.1182	0.8179	-100.2660
-0.5975	-0.2196	69.8200	0.8846	0.0854	0.7593	-100.2660
-0.545[illegible]	-0.3029	60.0712	0.8444	0.0423	0.6940	-100.2660
-0.4125	-0.3863	46.8726	0.8059	-0.0351	0.6033	-100.2660
-0.3333	-0.3863	40.7276	0.8059	-0.0709	0.5753	-100.2660
-0.2000	-0.3863	28.2243	0.8059	-0.1002	0.5524	-100.2660
-0.1000	-0.3863	15.0056	0.8059	-0.1132	0.5423	-100.2660
-0.0	-0.3863	0.0	0.8059	-0.1171	0.5332	-100.2660

PRESSURE DISTRIBUTION FOR STATION 11 Y= 7.5000

Y	Z	THETA	2-D POTENTIAL REAL	IMAG	1-D PATTERN MAG([illegible])	ARG([illegible])
-0.78[illegible]	-0.0	90.0000	1.0000	0.2680	0.7597	47.[illegible]
-0.[illegible]	-0.[illegible]	[illegible]	0.9585	[illegible]	0.71[illegible]	[illegible]
-0.7[illegible]	-0.[illegible]7	78.[illegible]	0.9111	0.2[illegible]	0.6761	47.[illegible]
-0.7[illegible]	-0.[illegible]	71.[illegible]	0.[illegible]	0.1[illegible]	0.[illegible]33	47.[illegible]
-0.7[illegible]	-0.[illegible]71	64.[illegible]1	0.9146	0.1131	0.5433	47.033[illegible]
-0.7[illegible]	-0.3[illegible]	64.[illegible]	0.[illegible]088	0.0[illegible]	0.[illegible]	47.03[illegible]
-0.7[illegible]	-0.[illegible]3	6[illegible]	0.[illegible]59	0.0[illegible]	0.[illegible]	47.[illegible]
-0.[illegible]	-0.[illegible]	52.[illegible]3	0.[illegible]	-0.0[illegible]	0.[illegible]	47.[illegible]
-0.[illegible]	-0.[illegible]3	3[illegible].[illegible]5	0.[illegible]59	-0.1[illegible]7	0.[illegible]	47.333[illegible]
-0.0	-0.[illegible]3	0.0	0.[illegible]59	-0.1[illegible]	0.3[illegible]	47.[illegible]332

[illegible] WAVE AMPLITUDE (INCIDENT+DIFFRACTED) [illegible]

[illegible]	MAGNITUDE	PHASE
0.0	1.0000	-90.0000
0.[illegible]	0.9176	-95.[illegible]
0.3[illegible]	0.[illegible]	-99.[illegible]
0.[illegible]	0.[illegible]	-[illegible]
1.[illegible]	0.[illegible]	-11[illegible]
1.[illegible]	0.[illegible]	-12[illegible]
2.[illegible]	0.[illegible]76	-152.[illegible]
3.[illegible]	0.[illegible]3	-174.[illegible]
4.[illegible]	0.[illegible]	139.[illegible]
[illegible]	0.[illegible]1	[illegible]
7.[illegible]	0.[illegible]7	[illegible]
[illegible]	0.[illegible]	[illegible].3247
10.[illegible]	0.7033	-4[illegible]
12.[illegible]	0.6934	-93.[illegible]
12.[illegible]	0.[illegible]	-117.[illegible]1
13.[illegible]	0.[illegible]	-140.[illegible]
13.[illegible]	0.[illegible]33	-151.[illegible]
14.[illegible]	0.[illegible]	-16[illegible]
14.[illegible]	0.5[illegible]	-174.[illegible]
15.[illegible]	0.[illegible]324	171.[illegible]

EXCITING FORCE DISTRIBUTION

X	REAL(F3)	IMAG(F3)	MAG(F3)	ARG(F3)
0.0	0.0	0.0	0.0	0.0
0.0[illegible]	0.[illegible]2436E-0[illegible]	-0.[illegible]+00	0.[illegible]	-[illegible]
0.1[illegible]	0.22698E-01	-0.14323[illegible]+01	0.14324[illegible]+01	-[illegible]
0.3[illegible]	0.51409E-01	-0.1[illegible]+01	0.13[illegible]+01	-[illegible]
0.[illegible]	0.[illegible]E-01	-0.1[illegible]+01	0.1251[illegible]+01	-[illegible]
0.7[illegible]	0.10[illegible]E+00	-0.1[illegible]+01	0.1[illegible]	-[illegible].265[illegible]
1.[illegible]	0.1[illegible]E+00	-0.11[illegible]+01	0.11[illegible]	-[illegible]
1.1[illegible]	0.1[illegible]25E+00	-0.11[illegible]+01	0.114[illegible]+01	-[illegible]
1.5[illegible]	0.15376E+00	-0.10[illegible]+01	0.10793[illegible]+01	-[illegible].813[illegible]
1.[illegible]	0.16[illegible]E+00	-0.1[illegible]+01	0.1[illegible]	-[illegible].71[illegible]
2.[illegible]	0.17[illegible]E+00	-0.10[illegible]+01	0.103[illegible]+01	-[illegible].0[illegible]
2.[illegible]	0.18[illegible]E+00	-0.1[illegible]+01	0.102[illegible]+01	-79.5611
3.[illegible]	0.1[illegible]777E+00	-0.[illegible]+00	0.9[illegible]+00	-78.5[illegible]
3.[illegible]	0.1[illegible]E+00	-0.[illegible]+00	0.9[illegible]+00	-78.[illegible]
3.7[illegible]	0.2127[illegible]E+00	-0.[illegible]+00	0.[illegible]+00	-77.357[illegible]
4.[illegible]	0.2247[illegible]E+00	-0.[illegible]+00	0.9[illegible]+00	-76.2[illegible]
5.1[illegible]	0.2337[illegible]E+00	-0.[illegible]+00	0.[illegible]821E+00	-75.[illegible]
6.[illegible]	0.2[illegible]E+00	-0.[illegible]+00	0.90[illegible]E+00	-7[illegible].[illegible]717
6.71[illegible]	0.2[illegible]E+00	-0.[illegible]+00	0.8[illegible]E+00	-73.72[illegible]
7.50[illegible]	0.25[illegible]E+00	-0.[illegible]+00	0.87[illegible]E+00	-7[illegible]
8.[illegible]	0.2[illegible]E+00	-0.[illegible]+00	0.857[illegible]E+00	-72.[illegible]711
9.[illegible]	0.2[illegible]E+00	-0.[illegible]+00	0.8438[illegible]E+00	-71.[illegible]

[illegible]	0.26932E+00	-0.79419E+00	0.82909E+00	-71.0531
[illegible]	0.27234E+00	-0.77132E+00	0.81767E+00	-70.5440
[illegible]	0.27524E+00	-0.75726E+00	0.80593E+00	-70.0277
[illegible]	0.27751E+00	-0.74516E+00	0.79591E+00	[illegible]
[illegible]	0.27780E+00	-0.73443E+00	0.78457E+00	-69.5257
[illegible]	0.27934E+00	-0.72532E+00	0.77715E+00	-69.2142
[illegible]	0.28004E+00	-0.73214E+00	0.76837E+00	[illegible]
13.07[illegible]	0.28375E+00	-0.72191E+00	0.73580E+00	[illegible]
13.5[illegible]	0.29354E+00	-0.72743E+00	0.71370E+00	-68.2044
13.87[illegible]	0.30603E+00	-0.74928E+00	0.80949E+00	-67.7011
13.9[illegible]	0.31404E+00	-0.76104E+00	0.82329E+00	-67.5765
14.25[illegible]	0.33151E+00	-0.77383E+00	0.84622E+00	-66.9367
14.375[illegible]	0.33509E+00	-0.77757E+00	0.84679E+00	-66.6953
14.625[illegible]	0.34293E+00	-0.78573E+00	0.85731E+00	-66.4214
14.9[illegible]	0.34709E+00	-0.81459E+00	0.88537E+00	-66.9299
15.0000	0.34479E+00	-0.85718E+00	0.92383E+00	-68.0934

TOTAL EXCITING FORCES AND B.M. AT X= 7.5000

	REAL	IMAG	MAG	PHASE
HEAVE=	-0.75349E-01	-0.49706E-01	0.90267E-01	-146.5879
PITCH=	-0.25822E-01	0.13182E-01	0.28992E-01	152.9561
B.M.=	-0.35655E-01	-0.14862E-01	0.39562E-01	-151.9195